

CLM Platform Implementation Manual

LK12-340

**Communication Link Module
(CLM)**

***CLM Platform
Implementation
Manual***

**LK12-340
Release Independent
03/2013**

Copyright, Notices, and Trademarks

Printed in U.S.A. – © Copyright 2013 by Honeywell International Srl.

Revision 09 – March 2013

While this information is presented in good faith and believed to be accurate, Honeywell disclaims the implied warranties of merchantability and fitness for a particular purpose and makes no express warranties except as may be stated in its written agreement with and for its customer.

In no event is Honeywell liable to anyone for any indirect, special or consequential damages. The information and specifications in this document are subject to change without notice.

TPS Network is a U.S. registered trademark of Honeywell International Inc.

Other brand or product names are trademarks of their respective owners.

Honeywell Process Solutions
1860 W. Rose Garden Lane
Phoenix, AZ 85027 USA
1-800 822-7673

About This Publication

This is the implementation manual for the Communication Link Module (CLM).

The CLM provides a standard communications platform, which allows the TPN to interface with a wide variety of devices/networks. With this interface, the operator has access to points resident on non-Honeywell devices that are connected to non-TPN networks. Access to these devices and networks is transparent so non-Honeywell devices appear to be directly connected to the Local Control Network (LCN).

There are two versions available for the CLM. Both are completely supported by this operations manual.

- **APP/CLM** installs to the APP node with an AM personality installed and emulates the previous CLMI/CCP communications boards used in the earlier version of the CLM. The concept of these boards and communications ports is carried over and emulated in the APP/CLM, however, the communications capability is now supported in the Windows environment and may include either physical or logical (emulated) communications devices. The actual device is transparent to the CLM operation. The user operation of the interface is the same.
Where there are references to hardware within this manual, such as: hardware, slots, CLMI/CCP boards, ports, device, etc., in the APP/CLM environment these items may be **physical** or **emulated**. Specifically for APP/CLM;
 - slots and boards are always **emulated**
 - ports and devices may be **physical or emulated** depending on the application
- **CLM** is the original CLM which is installed to an AM node and requires the CLM communications boards, CLMI or CCP. The user operation of the interface is the same.
Where there are references to hardware within this manual, such as: hardware, slots, boards, ports, device, etc., in the CLM environment these items are **physical**.

The CLM provides a standard communications platform, which allows the TPS and/or Experion Network to interface with a wide variety of devices/networks. With this interface, the operator has access to points resident on non-Honeywell devices that are connected to non-TPN networks. Access to these devices and networks is transparent so non-Honeywell devices appear to be directly connected to the TPN.

The complete CLM functionality is segmented into the following four logical layers:

- Process Application
- CLM Platform
- Network Interface
- Communication Protocol

The bottom three logical layers provide the communications foundation for the top layer or Process Application layer. The Process Application layer is the actual user application and is not considered part of the CLM Platform.

This *CLM Platform Implementation Manual* is primarily concerned with the second of these layers, the CLM Platform logical layer. This document is primarily intended for the engineer who will install, configure, and structure the TPS and/or Experion Network CLM Platform and its flexible interface with non-Honeywell devices that are connected to one of several possible non-TPN networks.

Important!	With CLM R200.1, CLM is supported only on Experion APP (Windows 2008 platform on both 32 bit and 64 bit variants). Note that CLM R200.1 is only meant for Windows 2008 nodes (Experion PKS R400/R410). CLM R200.1 does not work on Windows 2003.
-------------------	--

Support and Other Contacts

United States and Canada

Contact: Honeywell Solution Support Center
Phone: 1-800-822-7673
Calls are answered by dispatcher between 6:00 am and 4:00 pm Mountain Standard Time. Emergency calls outside normal working hours are received by an answering service and returned within one hour.
Facsimile: 1-973-455-5000
Mail: Honeywell TAC, MS L17
1860 W. Rose Garden Lane
Phoenix, AZ, 85027 USA

Europe, Middle East, and Africa (EMEA)

Contact: Honeywell TAC – EMEA
Phone: +32-2-728-2345
Facsimile: +32-2-728-2696
Mail: TAC-BE02
Hermes Plaza
Hermeslaan, 1H
B-1831 Diegem, Belgium
Email: tac-be02@honeywell.com

Pacific

Contact: Honeywell Global TAC – Pacific
Phone: 1300-364-822 (toll free within Australia)
+61-8-9362-9559 (outside Australia)
Facsimile: +61-8-9362-9564
Mail: Honeywell Limited Australia
5 Kitchener Way
Burswood, WA, 6100, Australia
Email: GTAC@honeywell.com

India

Contact: Honeywell Global TAC – India
Phone: +91-20- 6603-2718/19
1800-233-5051
Facsimile: +91-20- 6603-9800
Mail: Honeywell Automation India Ltd
56 and 57, Hadapsar Industrial Estate
Hadapsar, Pune –411 013, India
Email: Global-TAC-India@honeywell.com

Korea

Contact: Honeywell Global TAC – Korea
Phone: +82-80-782-2255 (toll free within Korea)
Facsimile: +82-2-792-9015
Mail: Honeywell Co., Ltd
4F, Sangam IT Tower B4-4 Block
1590, DMC Sangam-dong, Mapo-gu
Seoul, 121-835, Korea
Email: Global-TAC-Korea@honeywell.com

People's Republic of China

Contact: Honeywell Global TAC – China
Phone: +86- 21-2219-6888
800-820-0237
400-820-0386
Mail: Honeywell (China) Co., Ltd
33/F, Tower A, City Center, 100 Zunyi Rd.
Shanghai 200051, People's Republic of China
Email: Global-TAC-China@honeywell.com

Singapore

Contact: Honeywell Global TAC – South East Asia
Phone: +65-6823-2215
Facsimile: +65-6445-3033
Mail: Honeywell Private Limited
Honeywell Building
17, Changi Business Park Central 1
Singapore 486073
Email: GTAC-SEA@honeywell.com

Japan

Contact: Honeywell Global TAC – Japan
Facsimile: +81-3-6730-7228
Mail: Honeywell K.K.
New Pier Takeshiba, South Tower Building
20th Floor, 1-16-1 Kaigan, Minato-ku
Tokyo 105-0022, Japan
Email: Global-TAC-JapanJA25@honeywell.com

Elsewhere

Call your nearest Honeywell office.

World Wide Web

Honeywell Process Solutions website:

<https://www.honeywellprocess.com/>

Training Classes

Honeywell Automation College:

<http://www.automationcollege.com>

Table of Contents

SECTION 1 – CLM PLATFORM IMPLEMENTATION MANUAL	15
1.1 INTRODUCTION	15
1.2 CLM OVERVIEW	15
1.3 INTENDED AUDIENCE	16
1.4 WHAT YOU SHOULD READ	17
SECTION 2 –GENERAL PLATFORM OVERVIEW	19
2.1 INTRODUCTION	19
2.2 CLM SUPPORTING CAST	20
2.3 CLM LOGICAL STRUCTURE	23
SECTION 3 – CONCEPTS YOU NEED TO KNOW FIRST.....	25
3.1 TERMS AND RELATIONSHIPS	25
3.2 NETWORK DEFINITION TABLE	26
3.3 POINT INTERFACE TABLE STRUCTURE.....	27
3.4 BUILDING CLM INTERFACE STRUCTURES.....	29
SECTION 4 –INSTALLATION	31
4.1 INTRODUCTION	31
SECTION 5 –BUILDING CLM INTERFACE STRUCTURES.....	33
5.1 OVERVIEW	33
5.2 CLM CONFIGURATION BUILDER	34
5.3 CLM BUILDER PROCESSING MODES.....	36
5.4 BUILDER INPUT FILE.....	40
5.5 NETWORK DEFINITION TABLE BUILD ENTRY	43
5.6 POINT INTERFACE TABLE GROUP BUILD ENTRY.....	45
5.7 POINT INTERFACE TABLE BUILD ENTRY	46
5.8 POINT INTERFACE TABLE ENTRY BUILD ENTRY.....	47
5.9 BUILDER INPUT FILE EXAMPLES AND ANALYSIS.....	48
5.10 CLM CONFIGURATION BUILDER DISPLAYS.....	52
5.10 CLM CONFIGURATION BUILDER DISPLAYS, CONTINUED	53
5.11 BUILDER FUNCTION TARGETS.....	54
5.11 BUILDER FUNCTION TARGETS,.....	55
5.12 BUILDER DISPLAY MESSAGES AND PROMPTS	59
5.13 BUILDER STATUS MESSAGES	60
5.14 FILE MANAGER ERROR CODES.....	67
5.15 DATA ACCESS ERROR CODES	68
5.16 BUILDER DISPLAY ON-LINE HELP	69
SECTION 6 –INTERFACE STRUCTURE AND TDF	71
6.1 DEFINING INTERFACE STRUCTURE TEMPLATES	71
6.2 TEMPLATE DEFINITION FILE	72
6.3 TEMPLATE GENERAL ENTRY FORMAT AND RULES	73
6.4 NETWORK DEFINITION TABLE TEMPLATE.....	80
6.5 POINT INTERFACE TABLE GROUP TEMPLATE	90
6.6 POINT INTERFACE TABLE TEMPLATE.....	92
6.7 POINT INTERFACE TABLE ENTRY TEMPLATE	97
6.8 TEMPLATE DEFINITION FILE EXAMPLES	108

SECTION 7 – NETWORK INTERFACE DESIGN CONSIDERATIONS.....	117
7.1 INTERFACE DEFINITION AND USERS	117
7.2 NETWORK DESIGN AND CONFIGURATION STEPS	120
7.3 SYSTECH COM PORT EXTENDER.....	121
7.4 CLM STATUS DEFINITION FILE DESIGN	122
7.5 IMAGE POINT DATABASE DESIGN AND CONFIGURATION INFORMATION	125
7.6 IMAGE POINT OPERATION OVERVIEW.....	127
7.7 PIT INTERFACE DESIGN AND CONFIGURATION	131
SECTION 8 - IMAGE POINTS	133
8.1 STANDARD CLM POINT TYPES AND IMAGE POINT SELECTION	133
8.2 CLM PLATFORM STANDARD IMAGE POINT TYPES	138
8.2.1 SUPPORTED CLM IMAGE POINT TYPES.....	138
8.2.2 CLM DIGITAL IMAGE POINT TYPE	139
8.2.2.1 CLM DIGITAL POINT ALARM DEFINITION	144
8.2.2.2 CLM DIGITAL POINT PARAMETER DEFINITION	147
8.2.2.3 DIGITAL COMPOSITE POINT, 3-STATE LATCHED	156
8.2.2.4 DIGITAL COMPOSITE POINT, 3-STATE PULSED	165
8.2.2.5 DIGITAL INPUT POINT, 3-STATE.....	175
8.2.2.6 DIGITAL OUTPUT POINT, 3-STATE LATCHED	182
8.2.2.7 DIGITAL OUTPUT POINT, 3-STATE PULSED	191
8.2.2.8 DIGITAL COMPOSITE POINT, 4-STATE LATCHED	201
8.2.2.9 DIGITAL COMPOSITE POINT, 4-STATE PULSED	212
8.2.2.10 DIGITAL INPUT POINT, 4-STATE	223
8.2.2.11 DIGITAL OUTPUT POINT, 4-STATE LATCHED	231
8.2.2.12 DIGITAL OUTPUT POINT, 4-STATE PULSED.....	241
8.2.2.13 DIGITAL COMPOSITE POINT, 5-STATE LATCHED.....	251
8.2.2.14 DIGITAL COMPOSITE POINT, 5-STATE PULSED.....	262
8.2.2.15 DIGITAL OUTPUT POINT, 5-STATE LATCHED	273
8.2.2.16 DIGITAL OUTPUT POINT, 5-STATE PULSED.....	284
8.2.3 FLAG IMAGE POINT TYPE.....	295
8.2.4 NUMERIC IMAGE POINT TYPE.....	296
8.2.5 PULSE INPUT IMAGE POINT TYPE.....	297
8.3 CUSTOM CLM POINT TYPES.....	298
8.4 CL INTERFACES	299
8.4.1 CL INTERFACE TO NON-HONEYWELL DEVICE.....	300
8.4.2 CL INTERFACE TO NDT AND PIT STRUCTURES.....	310
8.4.3 CL INTERFACE TO IMAGE POINT DEMAND PROCESSING.....	315
SECTION 9 - CLM REDUNDANCY/AUTOMATIC FAILOVER.....	317
9.1 INTRODUCTION	317
9.2 CONFIGURATION/INSTALLATION OF CLM REDUNDANCY/AUTOMATIC FAILOVER.....	318
9.3 CLM REDUNDANCY	319
9.4 BUILD NDT ENTRY FOR CLM REDUNDANCY/AUTOMATIC FAILOVER.....	321
9.5 PERFORMANCE CONSIDERATIONS FOR OFF-NODE REQUESTS TO REDUNDANT CLMS	322
APPENDIX A –TERMINOLOGY/DEFINITIONS	325
A.1 INTRODUCTION	325
A.2 TERMINOLOGY	326
A.3 CLM VITAL STATISTICS.....	329
APPENDIX B - CLM PLATFORM RELEASE CONTENTS.....	331

B.1	INTRODUCTION	331
B.2	RELEASE MEDIA CONTENTS	332
APPENDIX C - CLM STARTUP		333
C.1	AUTOMATIC STARTUP AND INITIALIZATION	333
APPENDIX D - CLM MEMORY ACCESS UTILITY		335
D.1	INTRODUCTION	335
D.2	CLM MEMORY ACCESS SCHEMATIC	336
D.3	CLM MEMORY ACCESS ERROR CODES	345
APPENDIX E – UCLM: CCP AND UCIM REDESIGNED		347
INDEX		349

Tables

TABLE 1-1	SUGGESTED READING IN THIS MANUAL FOR DIFFERENT USERS	17
TABLE 2-1	CLM OVERVIEW	20
TABLE 2-2	FOUR LOGICAL LAYERS TO THE CLM.....	23
TABLE 5-1	BIF SPECIAL CHARACTERS	40
TABLE 5-2	NDT BUILD ENTRY ITEMS	43
TABLE 5-3	PITG BUILD ENTRY ITEMS	45
TABLE 5-4	PIT BUILD ENTRY ITEMS	46
TABLE 5-5	PITE BUILD ENTRY ITEMS.....	47
TABLE 5-6	ALPHABETICAL LISTING OF BUILDER STATUS MESSAGES	60
TABLE 5-7	COMMON FILE MANAGER ERROR CODES EXPLAINED.....	67
TABLE 5-8	COMMON DATA ACCESS ERROR CODES EXPLAINED.....	68
TABLE 5-9	ON-LINE HELP TOPICS	69
TABLE 6-1	TDF SPECIAL CHARACTERS	73
TABLE 6-2	NDT PARAMETER DEFINITIONS	82
TABLE 6-3	PITG PARAMETER DEFINITIONS.....	91
TABLE 6-4	PIT PARAMETER DEFINITIONS.....	94
TABLE 6-5	PITE PARAMETER DEFINITIONS	100
TABLE 7-1	DESIGN AND CONFIGURE A CLM NON-TPN NETWORK	120
TABLE 7-2	BUILDING IMAGE POINTS IN THE CLM.....	126
TABLE 7-3	POINT PARAMETER IDENTIFICATION PITE ENTRY OPTIONS.....	127
TABLE 7-4	PARAMETER CONFIGURATION PITE ENTRY OPTIONS.....	128
TABLE 8-1	CLM-SUPPORTED DATA POINT TYPES	133
TABLE 8-2	FUNCTIONS TO BE CONSIDERED WHEN SELECTING IMAGE POINTS	134
TABLE 8-3	DATA POINT TYPE OPTION DESCRIPTIONS AND CONSIDERATIONS.....	134
TABLE 8-4	TYPICAL FACEPLATE PARAMETERS.....	136
TABLE 8-5	TYPICAL FACEPLATE PARAMETERS FOR A SWITCH POINT	137
TABLE 8-6	SUPPORTED CLM IMAGE POINT TYPES.....	138
TABLE 8-7	MAX I/O CONFIGURATION FOR DIGITAL COMPOSITE	140
TABLE 8-8	DIGITAL POINT TYPE VS. I/O	141
TABLE 8-9	DIGITAL POINT ALARMING DEFINITION	144
TABLE 8-10	DIGITAL POINT PARAMETER DEFINITIONS	147
TABLE 8-11	3-STATE LATCHED DIGITAL COMPOSITE PARAMETER ENTRIES.....	159
TABLE 8-12	3-STATE PULSED DIGITAL COMPOSITE PARAMETER ENTRIES.....	168
TABLE 8-13	3-STATE DIGITAL INPUT PARAMETER ENTRIES	177
TABLE 8-14	3-STATE LATCHED DIGITAL OUTPUT PARAMETER ENTRIES.....	185
TABLE 8-15	3-STATE PULSED DIGITAL OUTPUT PARAMETER ENTRIES.....	194
TABLE 8-16	4-STATE LATCHED DIGITAL POINT PARAMETER ENTRIES	204
TABLE 8-17	4-STATE PULSED DIGITAL POINT PARAMETER ENTRIES.....	215
TABLE 8-18	4-STATE DIGITAL INPUT PARAMETER ENTRIES	225
TABLE 8-19	4-STATE LATCHED DIGITAL OUTPUT PARAMETER ENTRIES.....	234
TABLE 8-20	4-STATE PULSED DIGITAL OUTPUT PARAMETER ENTRIES.....	244
TABLE 8-21	5-STATE LATCHED DIGITAL COMPOSITE POINT PARAMETER ENTRIES.....	254
TABLE 8-22	5-STATE PULSED DIGITAL COMPOSITE POINT PARAMETER ENTRIES	265
TABLE 8-23	5-STATE LATCHED DIGITAL OUTPUT POINT PARAMETER ENTRIES.....	276
TABLE 8-24	5-STATE PULSED DIGITAL OUTPUT POINT PARAMETER ENTRIES.....	287
TABLE A.3-1	CONFIGURATION MAXIMUMS	329
TABLE B-1	CLM PLATFORM RELEASE MEDIA CONTENTS.....	332

Figures

FIGURE 1-1	READERS' "MAP" TO THIS MANUAL	18
FIGURE 2-1	COMMUNICATION LINK MODULE OVERVIEW	22
FIGURE 2-2	CLM LOGICAL STRUCTURE.....	24
FIGURE 3-1	SIMPLIFIED CLM DATA FLOW.....	25
FIGURE 3-2	POINT INTERFACE TABLE HIERARCHY	27
FIGURE 3-3	SIMPLIFIED INTERFACE STRUCTURE BUILD PROCESS OVERVIEW	29
FIGURE 5-1	BUILD PROCESS OVERVIEW	34
FIGURE 5-2	CONFIGURATION BUILDER DISPLAY LAYOUT (ABBREVIATED)	38
FIGURE 5-3	CONFIGURATION BUILDER DISPLAY LAYOUT	52
FIGURE 8-1	OPERATIONAL FLOW FOR 3-STATE LATCHED DIGITAL COMPOSITE	157
FIGURE 8-2	BUILD EXAMPLE FOR 3-STATE LATCHED DIGITAL COMPOSITE.....	161
FIGURE 8-3	OPERATIONAL FLOW FOR 3-STATE PULSED DIGITAL COMPOSITE	166
FIGURE 8-4	BUILD EXAMPLE FOR 3-STATE PULSED DIGITAL COMPOSITE.....	171
FIGURE 8-5	BUILD EXAMPLE FOR 3-STATE DIGITAL INPUT	178
FIGURE 8-6	OPERATIONAL FLOW FOR 3-STATE LATCHED DIGITAL OUTPUT.....	183
FIGURE 8-7	BUILD EXAMPLE FOR 3-STATE LATCHED DIGITAL OUTPUT	187
FIGURE 8-8	OPERATIONAL FLOW FOR 3-STATE PULSED DIGITAL OUTPUT	192
FIGURE 8-9	BUILD EXAMPLE FOR 3-STATE PULSED DIGITAL OUTPUT	197
FIGURE 8-9	BUILD EXAMPLE FOR 3-STATE PULSED DIGITAL OUTPUT, CONTINUED	198
FIGURE 8-10	OPERATIONAL FLOW FOR 4-STATE LATCHED DIGITAL COMPOSITE.....	202
FIGURE 8-11	BUILD EXAMPLE FOR 4-STATE LATCHED DIGITAL COMPOSITE.....	207
FIGURE 8-12	OPERATIONAL FLOW FOR 4-STATE PULSED DIGITAL COMPOSITE	213
FIGURE 8-13	BUILD EXAMPLE FOR 4-STATE PULSED DIGITAL COMPOSITE.....	218
FIGURE 8-14	BUILD EXAMPLE FOR 4-STATE DIGITAL INPUT.....	227
FIGURE 8-15	OPERATIONAL FLOW FOR 4-STATE LATCHED DIGITAL OUTPUT.....	232
FIGURE 8-16	BUILD EXAMPLE FOR 4-STATE LATCHED DIGITAL OUTPUT	237
FIGURE 8-17	OPERATIONAL FLOW FOR 4-STATE PULSED DIGITAL OUTPUT.....	242
FIGURE 8-18	BUILD EXAMPLE FOR 4-STATE PULSED DIGITAL OUTPUT	247
FIGURE 8-19	OPERATIONAL FLOW FOR 5-STATE LATCHED DIGITAL COMPOSITE.....	252
FIGURE 8-20	BUILD EXAMPLE FOR 5-STATE LATCHED DIGITAL COMPOSITE.....	257
FIGURE 8-21	OPERATIONAL FLOW FOR 5-STATE PULSED DIGITAL COMPOSITE	263
FIGURE 8-22	BUILD EXAMPLE FOR 5-STATE PULSED DIGITAL COMPOSITE.....	268
FIGURE 8-23	OPERATIONAL FLOW FOR 5-STATE LATCHED DIGITAL OUTPUT.....	274
FIGURE 8-24	BUILD EXAMPLE FOR 5-STATE LATCHED DIGITAL OUTPUT	279
FIGURE 8-25	OPERATIONAL FLOW FOR 5-STATE PULSED DIGITAL OUTPUT.....	285
FIGURE 8-26	BUILD EXAMPLE FOR 5-STATE PULSED DIGITAL OUTPUT	290
FIGURE D.2-1	CLM MEMORY ACCESS DISPLAY - CLM IDENTIFIER.....	337
FIGURE D.2-2	CLM MEMORY ACCESS DISPLAY - CLM BOARD IDENTIFIER.....	338
FIGURE D.2-3	CLM MEMORY ACCESS DISPLAY - STATUS.....	339
FIGURE D.2-4	CLM MEMORY ACCESS DISPLAY - READ DISPLAY	340
FIGURE D.2-5	CLM MEMORY ACCESS DISPLAY - FILE TARGET ENTRY PORT	342
FIGURE D.2-6	CLM MEMORY ACCESS DISPLAY - CURRENT STATE	343

Acronyms

AM	Application Module
BIF	Builder Input File
CCP	Communications and Control Processor
CDS	Custom Data Segment
CL	Control Language
CLM	Communication Link Module
CLMI	Communication Link Module Interface
DE	Digital Enhanced Communications (Smart Transmitter Protocol)
DEB	Data Entity Builder
ECP	Exception Change Processing
GUS	TPS Global User Station
HM	History Module
LCN	Local Control Network
NaN	A Floating Point Value Which is Not a Number
NCF	Network Configuration File
NDT	Network Definition Table
PIT	Point Interface Table
PITE	Point Interface Table Entry
PITG	Point Interface Table Group
PPS	Point Process Special
RB	Read Back
RBE	Report by Exception
STI	Smart Transmitter Interface
TDF	Template Definition File
TPN	TotalPlant Network
TPS	TotalPlant Solution System
UCIM	Universal Communications Interface Module Board
US	Universal Station
EAPP	Experion
APP	

Section 1 – CLM Platform Implementation Manual

1.1 Introduction

This manual

This document is the implementation manual for the TPS Network Communication Link Module (CLM) Platform, the generic foundation for interfacing non-TPN devices and networks with Honeywell's Local Control Network (LCN).

Scope

This document covers the CLM Platform layer.
It also covers part of the Network Interface layer—which is that portion of the interface that is oriented to the non-TPN network or device.
Details of the network and device-specific bottom two layers are documented separately, based on the type of network/devices being interfaced and the corresponding protocol.
In some cases, these two layers are interdependent and not separable.

1.2 CLM Overview

CLM Environment

The CLM provides the mechanism for the TPN to communicate with a wide variety of non-TPN networks. Two CLM environments are supported:

- **APP/CLM** is installed on an APP node with an AM personality installed and 'emulates' the previous CLMI/CCP hardware communications boards and slots used in the earlier version of the CLM. In this environment, ports and devices may be 'emulated' or physical and are application specific. The user operation of the interface is the same as the original CLM.
- **CLM** is the original CLM which is installed to an AM node and requires the CLM communications boards, CLMI / CCP, or UCIM.

The CLM modifies a standard Applications Module (AM) personality, with additional software. Additional software (and/or hardware) facilitates connection of the CLM to a non-Honeywell device or network (hereafter referred to as the non-Honeywell device or non-TPN network), resulting in a flexible communications interface.

The interface uses AM points with Custom Data Segments (CDS) to define image points in the CLM that represent points in the non-Honeywell network's database. Specific to the CLM application installed, the operator can monitor and change selected operating parameters in the remote point by changing the associated parameters in the CLM image point.

Note: With CLM R200.1, CLM is only supported on Experion APP (Windows 2008 platform on both 32 bit and 64 bit variants). Please note that CLM R200.1 is only meant for Windows 2008 nodes (Experion PKS R400/R410). CLM R200.1 does not work on Windows 2003.

Continued on next page

1.2 CLM Overview, Continued

Communication Link Module

The primary function of the CLM is to provide a standard communications platform with the flexibility for the TPS Network user to interface a wide variety of devices/networks to the LCN.

The CLM provides the operator a window of access to points located on a non-TPN network. The CLM interfaces with these remote points in a manner that simulates their direct connection to the LCN.

The complete CLM functionality is segmented into the following four logical layers:

1. Process Application
2. CLM Platform
3. Network Interface
4. Communication Protocol

The bottom three logical layers noted above provide the communications foundation for the upper Process Application layer—the actual user application, which is not considered part of the CLM Platform.

1.3 Intended Audience

Personnel involved

This document is designed for the use of all personnel involved in implementing the CLM including the following:

- Network Interface Design Engineers
- Installation Engineers
- Process Engineers

The document has been designed for convenient access by different functional users, moving from the most general overview to specific initial implementation details.

Refer to the following subsection for specific suggestions on reading.

1.4 What You Should Read

Suggested reading

Sections in this manual were designed to fill the needs of specific users. Depending on the level of understanding required, we recommend the sections indicated in Table 1-1.

Also refer to Figure 1-1, Readers' "Map" to This Manual, for a chart of the suggested reading programs and the required task order of CLM implementation.

Table 1-1 Suggested Reading in This Manual for Different Users

Reader/User	Manual Sections	Description
General Reader	1 through 3	These sections provide general introductory and overview information about the CLM Platform and are of interest to all readers.
Network Interface Design Engineer	1 through 6	The non-TPN network, including non-TPN devices and their corresponding addresses, will be defined by the Network Designer or Network Architect. This document includes the generic CLM Platform information required to aid in this network configuration definition.
Installation Engineer	1 through 4	Installation of the CLM requires a number of activities involving both hardware and software, these sections provide an overview. Specific installation instructions are included in the installation manual for either the APP/CLM (if on an APP node) or the CLM (if on an AM node). (Note: You will also have to reference additional documentation that is specific to the type of non-Honeywell network/devices and corresponding protocol).
Process Engineer	1 through 6	The Process Engineer defines detailed parameters required to support each non-TPN network controller point, point interface structures, and their relationships. This document includes the CLM Platform information required by the Process Engineer in preparing the non-Honeywell network/devices for interface with the user application.

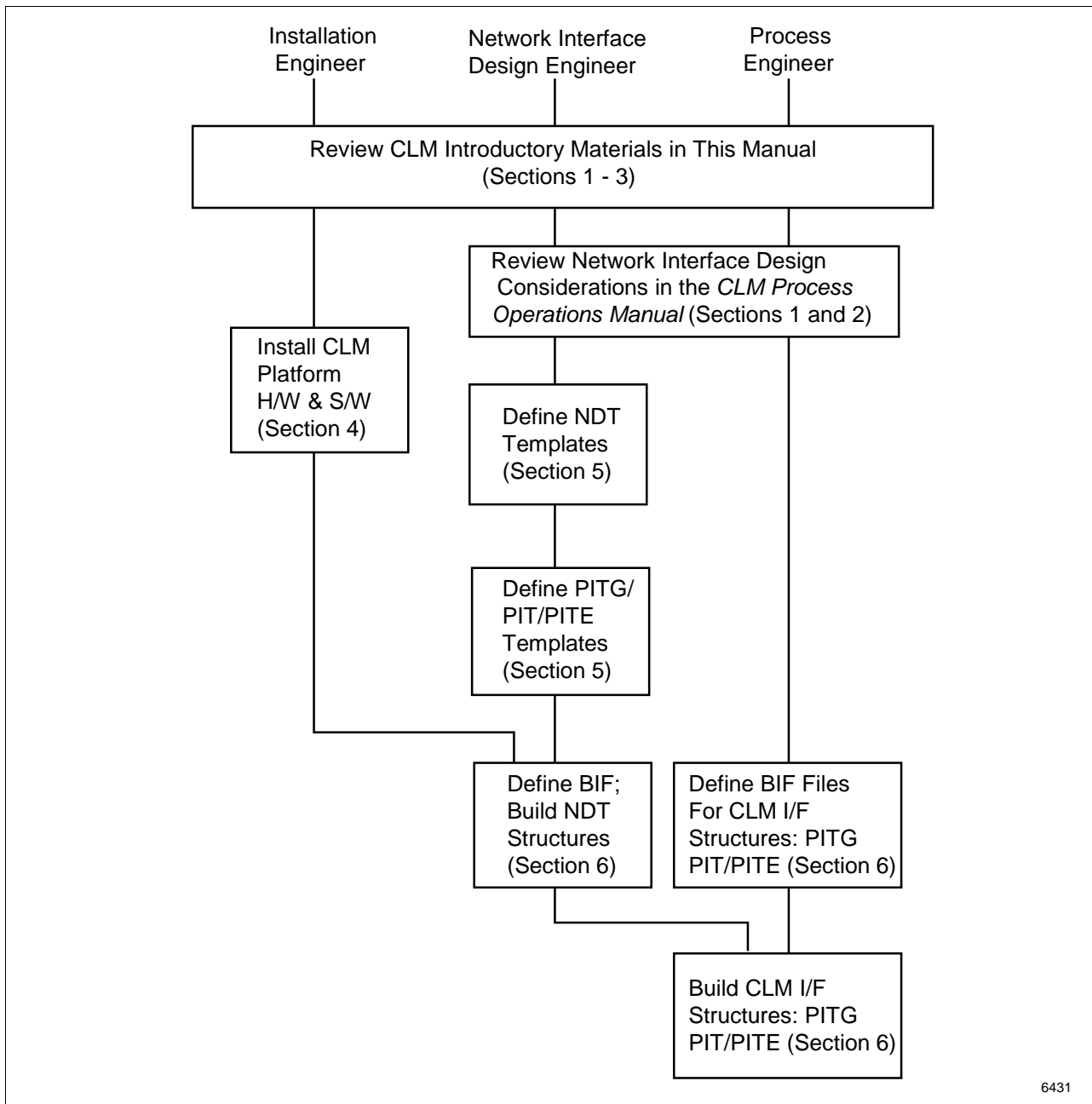
Continued on next page

1.4 What You Should Read, Continued

Readers' "map" to this manual

Figure 1-1 is a chart of the recommended reading plan described above. Also indicated is the required order of the main tasks involved in CLM implementation.

Figure 1-1 Readers' "Map" to This Manual



Section 2 –General Platform Overview

2.1 Introduction

CLM Platform

Two versions of the CLM Platform,

- **APP/CLM** is installed on an APP node with an AM personality installed and ‘emulates’ the previous CLMI/CCP hardware communications boards and slots used in the earlier version of the CLM. In this environment, ports and devices may be ‘emulated’ or physical and are application specific. The user operation of the interface is the same as the original CLM.
- **CLM** is the original CLM which is installed to an AM node and requires the CLM communications boards, CLMI / CCP, or UCIM.

It should be noted that the interface does support communications to other LCN-based node types (i.e., HG, NIM, etc.), but in a monitor-only basis. This means no direct outputs may be sent from these points.

This CLM interface is designed to fulfill a wide range of TPN and Experion user requirements for connecting non-TPN networks to the Honeywell TPS.

Note: With CLM R200.1, CLM is only supported on Experion APP (Windows 2008 platform on both 32 bit and 64 bit variants).

2.2 CLM Supporting Cast

APP/CLM (APP node) requirements

The APP/CLM Platform requires a TPS with the following items:

- User Station and History Module (HM)
 - APP with AM personality and additional network card
 - Windows specific CLM Communications-Oriented Software installation on APP
 - CLM Platform Displays
 - CLM User Network Configuration/Definition
-

CLM (AM node) requirements

The CLM Platform requires a TPN with the following items:

- User Station and History Module (HM)
 - Applications Module (AM)
 - Communication Link Module Interface (CLMI) Board
or
Communication and Control Process (CCP) Board
or
Universal Communication Interface Module (UCIM) Board
 - Communication Link Module Interface (CLMI)/CCP Paddleboard
 - CLM Communications-Oriented Software and Displays
 - CLM User Network Configuration/Definition
-

CLM overview

Figure 2-1, Communication Link Module Overview, in conjunction with the descriptions in Table 2-2, depicts the relationships among the various items that comprise a CLM.

Component description

The descriptions in Table 2-1 help explain the overview in Figure 2-1.

Table 2-1 CLM Overview

Component	Description
The AM personality configured for CLM	The AM personality (configured for use as a CLM) may be loaded either on an APP (APP/CLM) or an AM node (classis CLM). The node is loaded with CLM software that facilitates communications between the TPN and non-Honeywell devices on non-TPN networks. Characteristics of the user's network and devices are defined within CLM interface structures, referred to as "Network Memory," located within the AM.

Continued on next page

2.2 CLM Supporting Cast, Continued

Component
description,
continued

APP/CLM Emulated Board	The APP/CLM emulates the communications board within the Windows environment.
The CLMI Board	<p>The CLMI Board is a separate communications board that is connected to the CLM node.</p> <p>One or two CLMI boards, each with four ports, can be connected to a CLM, as required to establish each user's specific network configuration. Communications with non-Honeywell devices, on the user's non-TPN network, takes place by way of the CLMI ports. Communications oriented software, downloaded to the CLMI board, supplements the CLM software to provide the total functionality of the CLM.</p>
The CCP Board	<p>The CCP Board is a separate communications board that is connected to the CLM node.</p> <p>One or two CCP boards, each with four ports, can be connected to a CLM, as required to establish each user's specific network configuration. Communications with non-Honeywell devices, on the user's non-TPN network, takes place by way of the CCP ports. Communications oriented software, downloaded to the CCP board, supplements the CLM software to provide the total functionality of the CLM.</p>
The UCIM Board	<p>The UCIM Board is a separate communications board that is connected to the CLM node.</p> <p>One or two UCIM boards can be connected to a CLM, as required to establish each user's specific network configuration. Communications with non-Honeywell devices, on the user's non-TPN network, takes place via the UCIM ports. Communications software, downloaded to the UCIM board, supplements the CLM software to provide the total functionality of the CLM.</p>
Standard TPN Nodes	The standard TPN User Station and HM nodes are used by the CLM to accomplish the complete communications functionality. If the user's non-TPN network configuration warrants, multiple CLMs can be connected to the TPN.

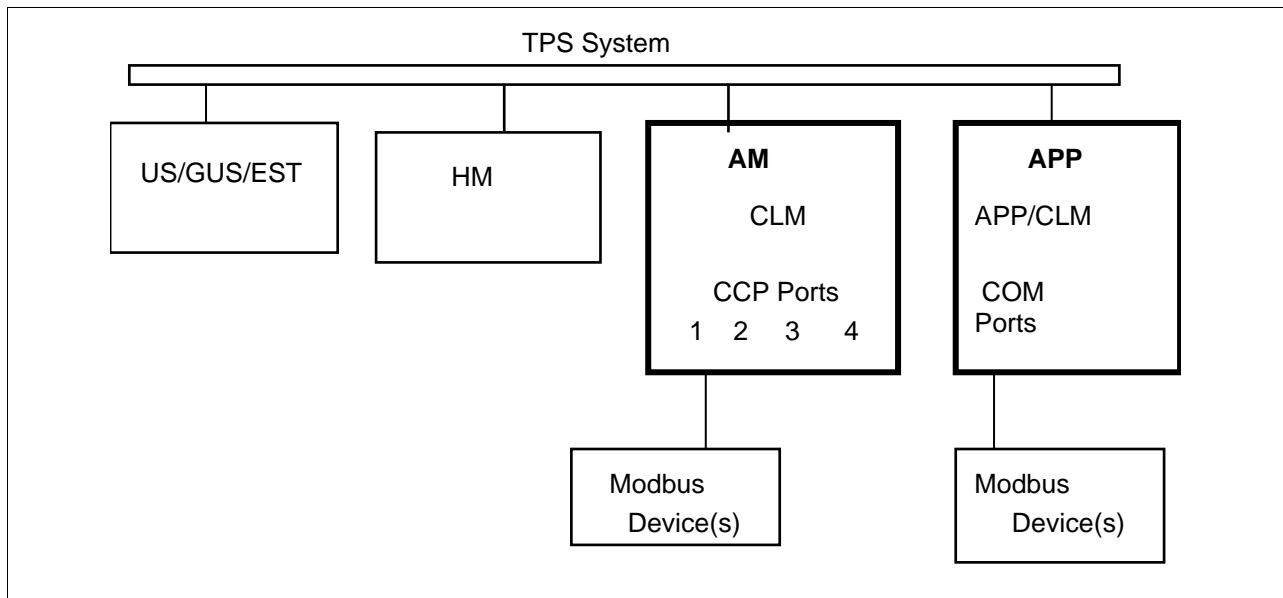
Continued on next page

2.2 CLM Supporting Cast, Continued

The CLM illustrated

Figure 2-1 provides a graphic overview of a Honeywell TPN with one APP/CLM and one standard CLM interface to non-TPN networks.

Figure 2-1 Communication Link Module Overview



2.3 CLM Logical Structure

Four logical layers

The CLM consists of four logical layers. The CLM interface supports the user application that is actually controlling a process that includes control points on the non-TPN network.

Table 2-2 gives a brief description of the four layers.

Table 2-2 Four Logical Layers to the CLM

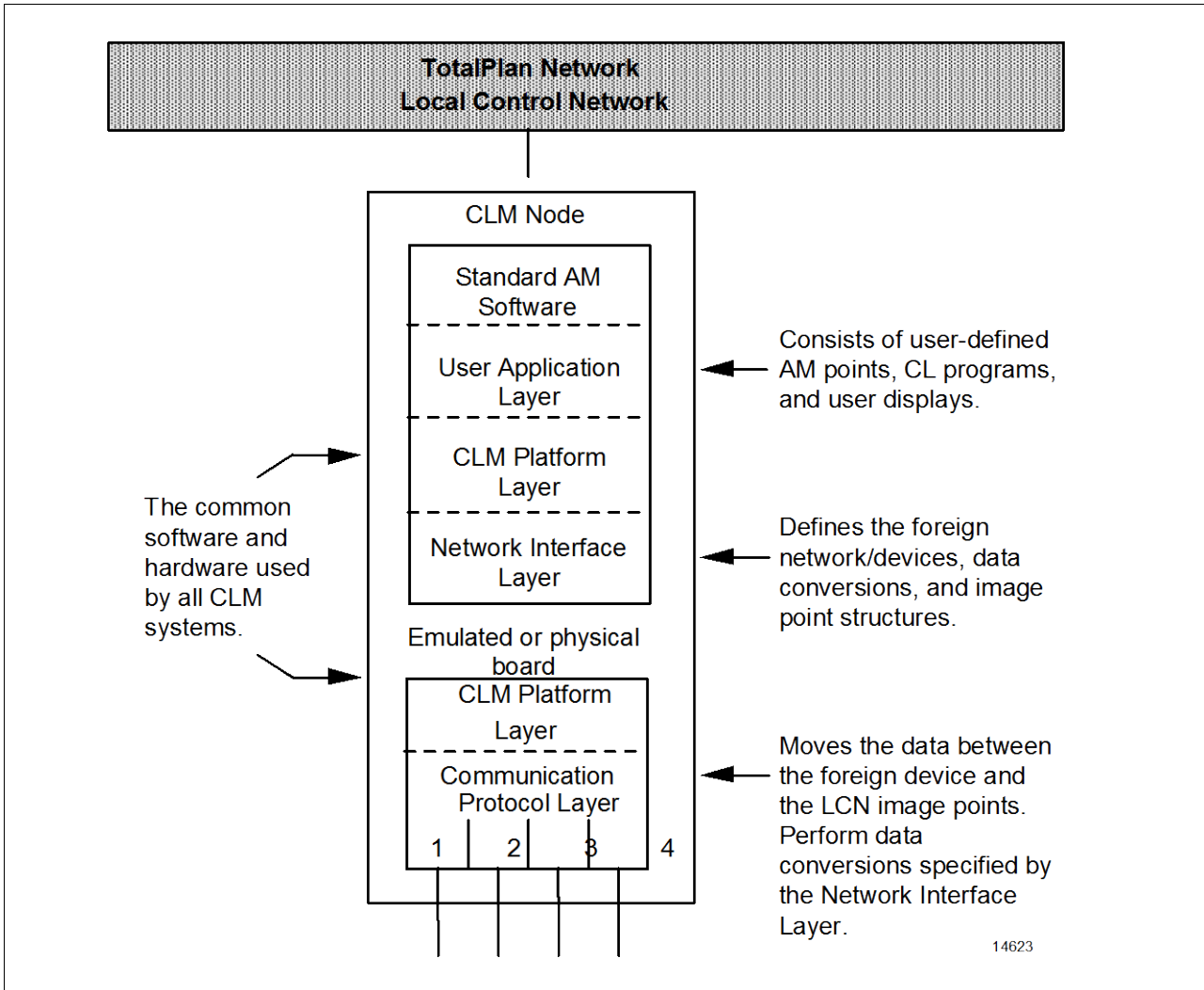
Layer	Name	Description
1	Process Application	<p>The process application layer represents the actual application that is monitoring and controlling a process.</p> <p>It consists of process application points, Control Language (CL) programs, and user displays that are built to monitor and control a process.</p> <p>The process application is built on top of the other three layers and is the end object of the CLM interface.</p>
2	CLM Platform	<p>The platform layer is the common layer that exists in all CLM environments and consists of both software and hardware.</p> <p>The platform provides a base for building a gateway to the TPN Local Control Network.</p> <p>The APP/CLM may have a physical or logical connection to the non-TPN network.</p> <p>The AM node physical connection is an RS232/485 link (or other supported link) between a CLMI /CCP/UCIM board in the CLM and a non-TPN network or device.</p> <p>This CLM Platform layer is the primary subject of this user's manual.</p>
3	Network Interface	<p>The network interface layer defines the type of non-TPN network and devices connected to it.</p> <p>This layer also includes the image points used to reflect the data on the non-TPN network and data conversion information.</p> <p>This user's manual includes information on the portion of this layer that is not oriented to a specific non-TPN network, depending on the user's environment.</p>
4	Communication Protocol	<p>The protocol layer is used to move the data between the non-Honeywell device control point and the CLM image point.</p> <p>All data conversions, as specified by the network interface layer, are performed by the communication protocol layer.</p>

Continued on next page

2.3 CLM Logical Structure, Continued

CLM logical structure Figure 2-2 depicts the relationships among the four layers.

Figure 2-2 CLM Logical Structure



Section 3 – Concepts You Need to Know First

3.1 Terms and Relationships

A brief review

This section includes a review of CLM interface structure terms and relationships that you should be familiar with prior to embarking on the definition of the interface structures that define your non-TPN network to the CLM.

Note that Appendix A covers terminology in greater detail.

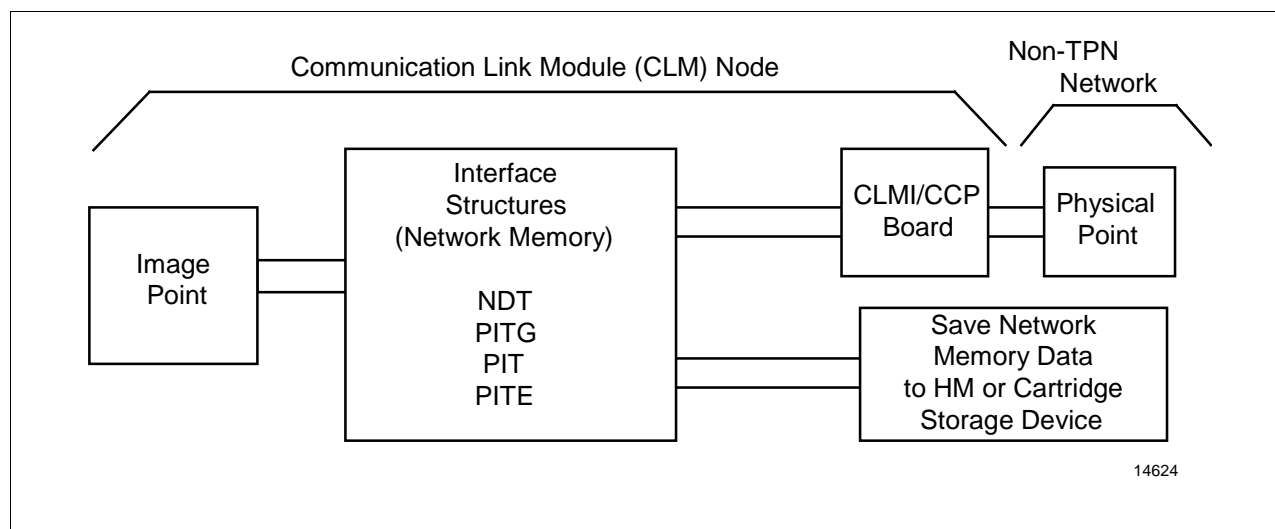
The objective of this section is to convey a basic understanding of the purpose of each structure, and how they relate to each other. If you are already acquainted with this structure terminology, you can bypass this section and go directly to the Defining Interface Structure Templates section.

The definition of all of these structures is accomplished by using the CLM Configuration Builder, which is part of the CLM Platform.

Note! For clarity, keep in mind that all references in this section to the CLMI/CCP boards may actually be the emulated boards when dealing with the APP/CLM.

Figure 3-1 depicts where the interface structures fit in the overall flow of data between a non-TPN network point and its corresponding image point, which is maintained by the CLM. In this example the board may be a physical or an emulated board.

Figure 3-1 Simplified CLM Data Flow



3.2 Network Definition Table

Component hierarchy

The Network Definition Table (NDT) contains the definition of every network component within the CLM environment including: the CLMI/CCP board(s), the ports on each board and each addressable component in the non-TPN network (i.e., gateway, bus, device, subdevice).

In the APP/CLM, the hierarchy of components is maintained even though the component may be physical or emulated.

In addition, the NDT contains the relationship between these network components, which is a hierarchy from the CLMI board, through its port(s), and through each addressable network component to the lowest level device address:

- CLM interface Board
 - CLM interface Board Port
 - Gateway
 - Device
 - Subdevice

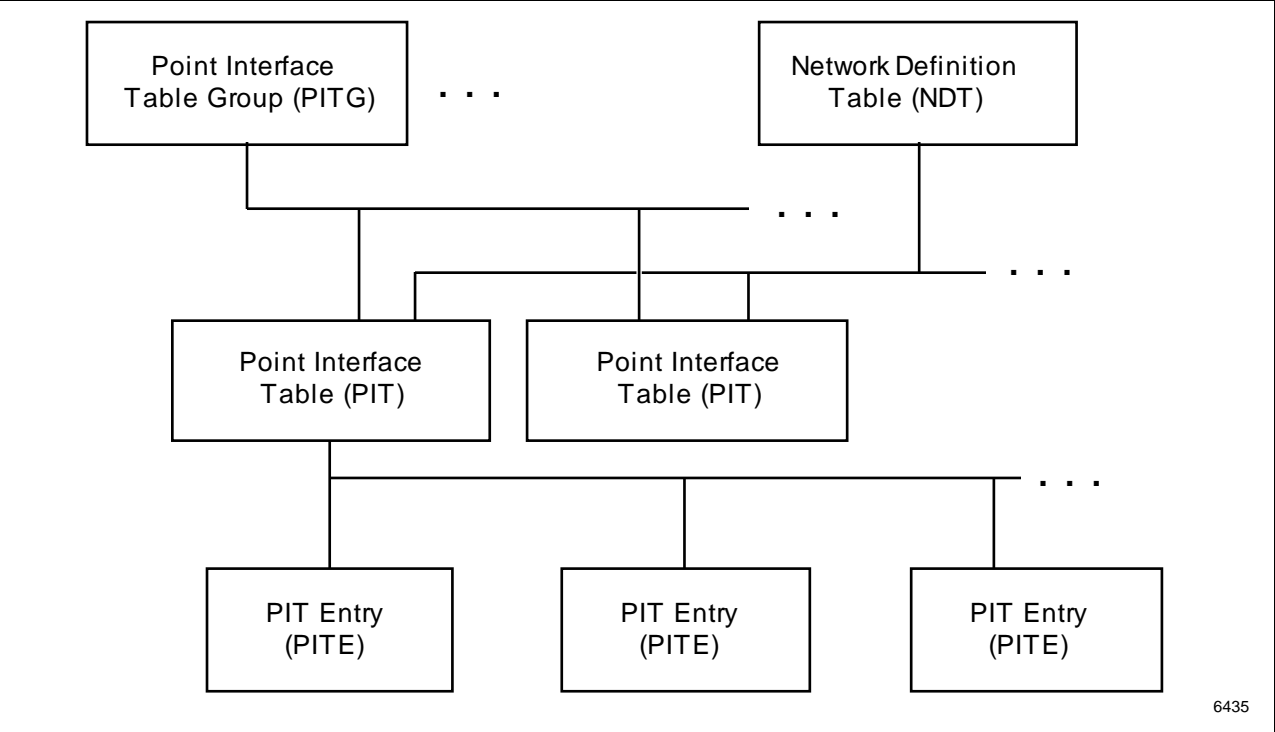
Each Point Interface Table (PIT) that represents a grouping of points by criteria that includes physical device address is also related to one NDT. The specific NDT is that which defines the addressable network component/device where each non-TPN point's data is accessed.

3.3 Point Interface Table Structure

Pivotal interface The Point Interface Table structure represents the primary interface between the CLM and the non-TPN network. This consists of the Point Interface Table (PIT) and the subordinate Point Interface Table Entry (PITE). The PIT can *optionally* be related within a Point Interface Table Group (PITG).

Point interface table structure Figure 3-2 depicts the relationship between these three interface structures (PIT, PITE, and PITG).

Figure 3-2 Point Interface Table Hierarchy



Continued on next page

3.3 Point Interface Table Structure, Continued

Point Interface Table Group (PITG)

A Point Interface Table Group optionally defines a *set* of Point Interface Tables (interfacing with the same non-Honeywell network/port) that are to be processed as a group. This is helpful when data collection across different devices needs to be synchronized.

A PITG represents the highest level entity in the overall point interface hierarchy.

Note: If the PITG structure is used, it must be defined prior to the definition of member PITs within the group. A maximum of 50 Point Interface Tables can be grouped under one PITG, for optimizing data acquisition as required.

Point Interface Table (PIT)

The Point Interface Table (PIT) structures contain the relationships between CLM image points and their corresponding actual non-TPN network points, along with conversion and acquisition information.

Each point in a PIT is represented by a Point Interface Table Entry (PITE). Each PIT defines a set of points grouped together, based on both processing time and physical device address. Each PIT must be related to the NDT that defines the addressable network component/device where the PIT's set of points is located. A PIT can also optionally be a member in a Point Interface Table Group (PITG).

To build a PIT, you must first define the PIT, and then add PIT's to the table. If a PIT is to be assigned to a PIT Group, then the PITG must be defined first.

The criteria for grouping non-TPN network points into specific Point Interface Tables is based on both desired processing frequency and physical device address. Therefore, the total number of points associated with one physical device address may be grouped into multiple Point Interface Tables, with each PIT having its own processing cycle.

All points defined within a specific PIT must have the same physical device address, and the same processing cycle. However, a device can have several PITs, as noted above.

3.4 Building CLM Interface Structures

Point Interface Table Entry (PITE)

The Point Interface Table Entry (PITE) defines one non-TPN network point within a PIT. The PITE is referenced by the CLM to determine interface input and output requirements between the CLM image point and the non-TPN network point.

Each PITE within a PIT must have the same processing cycle and the same physical device address.

Each Point Interface Table Entry (PITE) consists of one or more element definitions that define the characteristics of the non-TPN network point to the CLM. These elements include relating the point's physical device address to its corresponding CLM image point, point processing options, and a set of parameters that define the point's input and output requirements. Each non-TPN network point must be defined in one PITE.

Note: An LCN point can appear in only one PIT.

The parameter is the lowest element in the PIT structure.

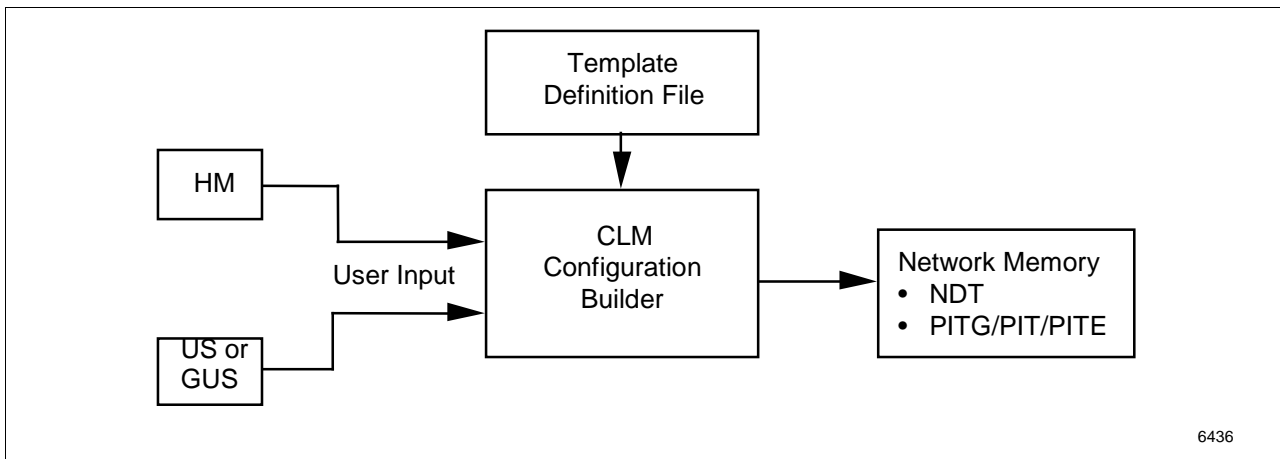
Overview

All of the CLM interface structures reviewed in this section are defined using the CLM Configuration Builder (referred to as the Builder), which is part of the CLM Platform.

Interface structure build process

Figure 3-3 shows an overview of the interface structure build process.

Figure 3-3 Simplified Interface Structure Build Process Overview



Continued on next page

3.4 Building CLM Interface Structures, Continued

Build Summary

- Define a template file for the type of interface structure required. Store the Template Definition File (TDF) for the interface structure type in the HM.
Template Definition File skeletons for each interface structure type are normally supplied by Honeywell. They may require modification based on your specific requirements.
- Define the Builder Input File (BIF) source statements, representing the definition of the interface structure.
The BIF is a set of ASCII input forms made up from the Template Definition Files, for input to the CLM Builder function. BIF statements orient a generic interface structure type to your requirements for a specific network component or element.
- Input BIF to the Build Process, Figure 3-3, using the CLM Configuration Builder.
Note: The Build Process accepts either file input or on-line input from the TPS Station.
- Build Process – performed by CLM Configuration Builder:
Validate the structure definition input based on the corresponding TDF in the HM for the interface structure type.
Expand structure definition, also based on a predefined template, to derive the complete structure definition (including nonvisible segments).
Store the interface structure in Network Memory, a segment of the CLM/AM memory.
- CLM interface structure is now available for use by the CLM Platform.

Note: Although the HM is the preferred storage medium, TPN removable media can alternately be used as Builder Input File or Template File source.

Section 4 –Installation

4.1 Introduction

CLM Versions

There are two versions available for CLM Platform Software installation. Select the one that applies to your system configuration.

- **APP/CLM Platform** installs to the APP node with an AM personality installed and emulates the previous CLMI/CCP communications boards used in the earlier version of the CLM.

Refer to the *TPS/CLM Planning, Installation Manual LK02-200*.

- **CLM Platform** is the original CLM, which is installed to an AM node and requires the CLM communications boards, CLMI, CCP, or UCIM.

Refer to the *CLM Planning, Installation, and Service Manual LK02-100*.

Note: With CLM R200.1, CLM is only supported on Experion APP (Windows 2008 platform on both 32 bit and 64 bit variants).

When the software and hardware installation has been completed, continue with Section 5 – Build CLM Interface Structures

STOP!

The installation of the CLM Platform software must have been completed before continuing beyond this point! You should have referred to one of the installation manuals referenced above for detailed installation instructions.

Section 5 –Building CLM Interface Structures

5.1 Overview

Defined by Builder

The following CLM interface structures are defined using the CLM Configuration Builder, which is part of the CLM Platform:

- Network Definition Table (NDT)
- Point Interface Table Group (PITG)
- Point Interface Table (PIT)
- Point interface Table Entry (PITE)

If these terms are unfamiliar to you, please review Section 3, Concepts You Need to Know First, as well as Appendix A, Terminology/Definitions, both in this manual.

5.2 CLM Configuration Builder

Principles

The CLM Configuration Builder (referred to as the Builder) is an Engineering-level function.

The Builder is used to generate the CLM interface structures required to support your non-Honeywell network and devices.

The Builder validates each structure definition entry submitted, based on a pre-defined Template Definition File (TDF). Therefore, all TDFs for the application must be completed and available to the Builder in the HM or on removable media at an LCN station prior to building the CLM interface structures.

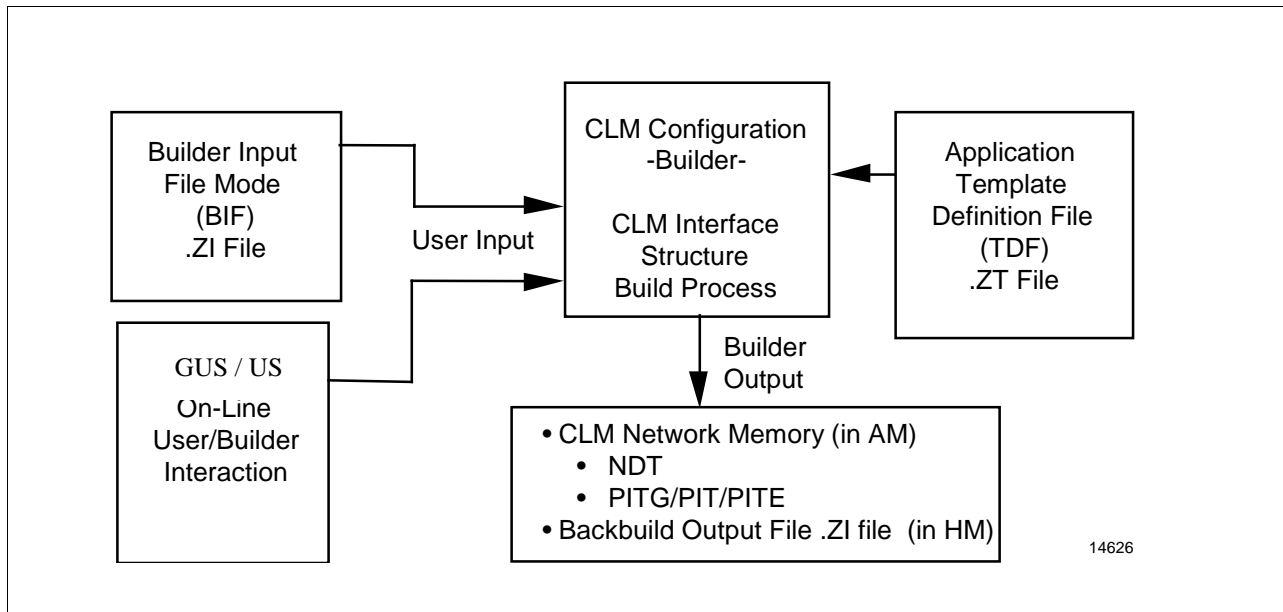
The Builder is called from the **CLM BUILDER** target on the CLM Network Status display.

The Builder accepts input definitions from either a file or on-line from the User Station. There is also a backbuild feature, where the Builder generates input definitions from the CLM interface structures that currently exist in Network Memory.

Figure 5-1 gives a graphic overview of the Build process.

Build process overview

Figure 5-1 Build Process Overview



Continued on next page

5.2 CLM Configuration Builder, Continued

The TDFs

Each Template Definition File (TDF) provides the complete definition of an interface structure entry type to the Builder, including:

- its format,
- constants for the entry type, and
- required user inputs with corresponding validation criteria.

For example, an NDT file must be defined for each different *type* of addressable component within the network definition (i.e., one for CLMIs, one for Ports, etc.).

The Builder uses the first entry of the TDF as the file name (for example, ENTRY = MP_PORT) and scans the configured media directory for the file. Once located, the Builder uses data within the matching template file to validate your input.

The Builder also uses information in the TDF that relates to the specific entry type to complete the structure definition.

Refer to Section 6, Defining Interface Structure Templates, for an explanation on building TDFs.

5.3 CLM Builder Processing Modes

Introduction

There are three CLM Builder processing modes:

- Input File mode
- On-line Builder mode
- Backbuild mode

When a CLM node is operational, the CLM Builder can be accessed by touching a target on the initial CLM Network Status display. Either the input file mode or the on-line interactive mode can be used.

Input File mode

The Input File mode is primarily used to build the initial CLM interface structure definitions required for your network from a user-edited file. The active process tells you if an entry already exists, in which case you have the option of either replacing it with the entry from the input file or leaving the existing entry unchanged. At your convenience, you can define CLM interface structures off-line and process them later through the Input File mode.

Selecting this mode

Selecting the **FILE INPUT** function target on the initial CLM Configuration Builder Display (Figure 5-2) informs the Builder that you want to input a file to the Build process – rather than working on-line in ENTRY TYPE MODE or Backbuilding in BACKBUILD MODE.

The system prompts you to enter the file pathname. Then it searches for the file on the specified media. Once the file is located, the Builder prompts you to select the **BUILD** function target to start the process.

Files used as input to the Builder by the Input File Mode are referred to as Builder Input Files (BIFs). BIFs are covered in detail in the following section.

Input data factors

When processing in the Input File mode, the Builder displays input data errors for your review and on-line corrective action.

The following alternatives are available to you if the Builder displays an error in the Input File mode. User action is required to continue file processing:

- Edit/correct the error noted and continue the Build process from the current entry.
 - Skip the entry in error (current entry) and request the Builder to continue with the next entry in the Builder Input File. Generally, this allows the Builder to process the entire BIF. You can then manually enter the bypassed structure definitions, in the on-line mode.
 - Break (or abort) the current Builder process after the current entry. All interface structure definitions that were successfully validated, and stored in Network Memory by the Builder up to the requested Break point remain in Network Memory. If only minor additions and/or deletions are required to complete the interrupted Builder process, it may make sense to finish using the on-line mode.
-

Continued on next page

5.3 CLM Builder Processing Modes, Continued

On-Line Builder mode (Single Entry)

In the on-line mode, all Builder functions are available to you at the User Station, as follows:

- Add,
 - Modify,
 - Delete, or
 - View a CLM interface structure entry.
-

Selecting this mode

Selecting the ENTRY TYPE function target, on the initial CLM Configuration Builder Display (Figure 5-2), informs the Builder you want to work in this mode. The system prompts you for the Entry Type.

The Builder uses your input to find the file name of the Template Definition File (TDF) for a NDT, PITG, PIT, or PITE interface structure.

If the Builder locates the template, follow the Builder's prompts to modify the interface structure, delete it, or add a new one.

If the template is not found, the Builder displays the error, and you can either input a corrected Entry Type or select another function.

Continued on next page

5.3 CLM Builder Processing Modes, Continued

More information on this mode

The following CLM Configuration Builder Displays section has more information on working with the Builder in the On-Line Builder mode.

Figure 5-2 illustrates the top of the CLM Configuration Builder display.

Figure 5-2 Configuration Builder Display Layout (Abbreviated)

The diagram illustrates the top portion of the CLM Configuration Builder display. It features a header section with the title "CLM CONFIGURATION BUILDER" and a sub-header "> Select Builder Option". Below this, there are several input fields and status indicators. On the left, there is a vertical stack of three boxes, with the top one labeled "ENTRY TYPE". In the center, there is a larger box containing the text "CLM ID: 55", "ENTRY TYPE:", "DET STATUS", and "STATUS : OK". To the right of this central box, there is another vertical stack of three boxes, with the top one labeled "FILE INPUT" and the middle one labeled "BACKBUILD". Below these fields, there is a horizontal bar with the labels "ENTRY DATA", "CLR", and "BUILD DATE". At the bottom right of the main display area, the number "6438" is displayed.

CLM CONFIGURATION BUILDER			
> Select Builder Option			
ENTRY TYPE	CLM ID: 55	FILE INPUT	
	ENTRY TYPE:	BACKBUILD	
	DET STATUS		
	STATUS : OK		
ENTRY DATA CLR BUILD DATE			

6438

Continued on next page

5.3 CLM Builder Processing Modes, Continued

Backbuild mode

The Backbuild option generates a Backbuild source file, in Builder Input File (BIF) format, using the current CLM interface structures existing in Network Memory as input definitions to the process.

The Backbuild option is a convenient mechanism to generate a BIF based on the current interface structures in Network Memory. The result of this process is a Builder Input File containing either ALL Network Memory entries for up to two boards (All releases) or a finite set of entries selected in a top-down hierarchy from a user entered NDT or PIT name (R360.3 and later ONLY).

Selecting this mode

Selection of the **BACKBUILD** function target on the initial CLM Configuration Builder display (Figure 5-2) requests the Builder to initiate a backbuild process.

The process responds with a request to enter the file pathname, where the Builder will store the results of the Backbuild.

WARNING!

If the file name entered already exists, the Builder will display a warning to the user. Selecting the **BUILD** target at this point will overwrite the contents of the named file.

Backbuild Processing Options (R360.3+ only)

The process will then display a request to enter the search parameters for the backbuild. These options are:

- ALL = Backbuild all Network Memory entries (includes both boards, if applicable)
- NDT = Backbuild all Network Memory entries *at and below* the user-entered NDT name.
- PIT = Backbuild all Network Memory entries *at and below* the user-entered PIT name.

After selecting a backbuild processing option, the process locates the requested file or creates a new file and then if applicable, validates any entries entered by the user to qualify the backbuild.

The Builder then displays a request to select the **BUILD** function target to begin the backbuild processing.

5.4 Builder Input File

BIF definition

Files used as input to the Builder in the Input File mode are referred to as Builder Input Files (BIFs).

A BIF contains interface structure definition entries.

A BIF consists of a set of source statements, each of which can use the full 80-character line length.

Introduction

This subsection describes the BIF source format, general rules for BIF source statements, and specific entry definitions required to define the various CLM interface structures.

BIF common source format

While the specific structure definition entries in the BIF and their corresponding template files may vary by application, the following general entry format and entry definitions remains the same.

- Within a BIF source line, the special characters in Table 5-2 are recognized by the Builder.
- Each BIF source input statement begins with an item name.
- Each name under the "Item Name" heading in Tables 5-3 through 5-6 represents a specific literal value entry that must have a corresponding entry in the TDF.
- The Builder uses the corresponding TDF definitions to validate the BIF format source input statements.

Special characters

The Builder recognizes special characters and their use as listed in Table 5-1.

Table 5-1 BIF Special Characters

Special Character	Definition
*	<u>Comment line</u> : You can use an asterisk for commentary anyplace on the line. The Builder terminates processing of the line when an asterisk is encountered.
=	<u>User Visible entry</u> : This sign indicates to the Builder that an equivalence must be interpreted and assigned to a particular definition parameter. User input is required through the Builder display.
;	<u>Entry separator</u> : The semicolon is used to separate multiple equivalence entries on the same source line. If there is only one entry on a line, the semicolon is not required.
:	<u>Internal entry</u> : This sign indicates to the Builder that an equivalence must be interpreted and assigned to a particular definition parameter. This entry is not user visible through the Builder display and gets its input from the TDF or, if the entry already exists, from Network memory. <i>(Not Used in BIF)</i>

Continued on next page

5.4 Builder Input File, Continued

BIF format

Each BIF source statement is in the following general format:

< ITEM > < = > < VALUE >

where:

<ITEM>	<u>Name of the entry</u> : A maximum of eight alphanumeric characters are allowed. This name must have a matching entry in the TDF name.
<=>	<u>Equal sign</u> : This sign indicates to the Builder that an equivalence must be assigned for the defined value.
<VALUE>	<u>User input</u> : The VALUE field represents the desired input value to satisfy the definition of the parameter. The input value must meet the conditions expressed in the corresponding template file parameter definition.

BIF usage rules

Rules for entering source statements in the BIF are summarized below:

- Each BIF file name must have a "ZI" extension (i.e., BIF_FILE.ZI).
 - BIF files can reside on an HM or removable media mounted at an LCN station (for subsequent input to the Builder using the file input mode).
- Note:** The directory is open to the user.
- Each set of BIF source statements that define an interface structure, must begin with the following source statement: ENTRY = <VALUE> or DELETE = <VALUE>, where <VALUE> is the name of a corresponding TDF. *Also refer to the section on deletions.*
 - Each set of BIF source statements must complete with an END statement.
 - Comments begin with an asterisk (*) and can be anywhere on the line; Builder processing of the source input statement ends at that point.
 - BIF files are an ordered set of statements. Blank characters may be used to improve readability, but cannot be used within the item name or value fields.
 - Each statement begins with a 1-8 character name that must match a corresponding entry in the related template (same name as ENTRY field value).
 - Sequence of BIF source statements and TDF user-visible definitions must be the same.
 - Each BIF source statement must be contained within the same line of the file.
 - Multiple BIF source statements can be present on the same line, but must be separated by a semicolon (;).
 - An interface structure must be defined in the BIF (or already exist in Network Memory) before it can be referenced by another interface structure definition (i.e., A PIT that references a specific NDT must have had that NDT defined previously in the file or have already been built through the Builder process).

Continued on next page

5.4 Builder Input File, Continued

Deletions

The BIF file can be used to delete existing entries as well as add new entries. This is accomplished simply by replacing the “ENTRY =” literal in the specific entry with the literal “DELETE =”, saving the file and executing the file from the Builder as normal FILE INPUT. The Builder will ignore attempts to delete entries that do not exist.

Rules for creating a BIF for deleting entries are summarized below:

- Literal “ENTRY =” replaced by “DELETE =”.
 - Entries must be in bottom-up order (i.e., PITE, PIT, PITG, NDT)
 - All Delete entries must appear prior to their corresponding ADD entries.
 - All PITE field user-visible entries between the PIT_NAME and END fields are ignored in the DELETE process.
 - All other standard BIF rules apply.
-

5.5 Network Definition Table Build Entry

Table 5-2 defines the user input to create a Network Definition Table (NDT) data structure.

Table 5-2 NDT Build Entry Items

Item Name	Definition
ENTRY = <VALUE>	<u>NDT type</u> : This value corresponds to an NDT-type TDF. Therefore, this entry indicates to the Builder that the input that follows represents a specific NDT entry (i.e., CLMI, PORT, DEVICE, etc.). The entered value must be an NDT-type TDF name.
NAME = <VALUE>	<u>Component name</u> : This entry is the user-defined name of the device or network component. The entered value must be from 1 to 8 characters, and the name must be unique across all components in the network!
A1 = E1 A2 = E2 A3 = E3 A4 = E4 A5 = E5 A6 = E6 A7 = E7 A8 = E8	<u>Address entry</u> : This entry is the NDT or device address for the component. The entered value is the total address, from the top level addressable network component (i.e. board) down to this addressable device (i.e., if device = first level controller, address would be of the form): A1_CLMI=1; A2_PORT=2; A3_GATEWAY=2; A4_CONTLR=5. The complete address entry form is: A1aaa=E1; A2aaa=E2; . . . A8aaa=E8 where: A _n aaa = Address element mnemonic names of the addressable component. <ul style="list-style-type: none"> All 8 elements are not generally required to define the component address. However, address elements not used to define the component must be initialized to -1. E _n = Integer address of the component, for the corresponding address element of the component's total address. Note : A1 is always the CLM Interface Board address element; E1 is the CLM Interface Board number. A2 is always the Port address element; E2 is the Port number. Use A3 through A8 address elements as required to complete the total address for the subject component.
"xxxx" = <VALUE>	<u>Application specific parameter</u> : The "xxxx" symbol represents a 1 to 8 character value that corresponds to an equivalent template definition in the TDF. Use of variable parameters in a specific application's NDT entry type is dependent upon the non-TPN network being interfaced to the CLM. When additional user input is required for a specific NDT entry type, this parameter is mandatory. User input is defined in the corresponding TDF definition.

Continued on next page

5.5 Network Definition Table Build Entry, Continued

NDT build entry table, continued

Table 5-2 NDT Build Entry Items (Continued)

Item Name	Definition
Note1: The NDT Template for a PORT on a CLM Interface Board requires these six additional key parameters.	
Note2: The NDT templates for the port and all devices are part of the application interface package and are listed here for information purposes only.	
1. BAUD = <VALUE>	<u>Baud rate code</u> : Baud rate for the communication port or channel. Default is 4. Valid values and their corresponding meanings are: 0 = 300 1 = 1200 2 = 2400 3 = 4800 4 = 9600 (default) 5 = 19200 6 = 38400
2. DBITS = <VALUE>	<u>Data bits number</u> : Number of data bits in a communication byte. Valid values are 7 or 8. Default is 8.
3. SBITS = <VALUE>	<u>Stop bits code</u> : Integer value that represents the number of stop bits used to denote the end of a transmitted data byte. Default is 1. Valid values and their corresponding meanings, are: 1 = 1 stop bit (default) 2 = 2 stop bits 3 = 1.5 stop bits
4. PARITY = <VALUE>	<u>Data parity code</u> : Integer value that represents the data parity option for transmitted data bytes. Default is 0. Valid values, and their corresponding meanings are: 0 = No parity (default) 1 = Odd parity 2 = Even parity
5. RETRY = <VALUE>	<u>Retry code</u> : Integer value that defines the number of retries to attempt on a communication failure before aborting the request. Default is 2; Default of 2 means a total of three tries are attempted before aborting the request.
6. TIMEOUT = <VALUE>	<u>Timeout period</u> : Integer value that defines the base timeout period (application dependent) for a no-response communication failure before the request is retried. A default may have been defined in the port's NDT template by your Network Interface Designer.
7. MODEM = <VALUE>	<u>MODEM ENABLE</u> : This value enables the RTS and CTS signals required for modem communications. Also REQUIRED if RS 485 communications is configured. Y = ENABLE, N = DISABLE
END	<u>End of entry</u> : This defines the end of the NDT entry data.

5.6 Point Interface Table Group Build Entry

PITG build entry table Table 5-3 defines the user input to create a Point Interface Table Group (PITG) interface structure.

Table 5-3 PITG Build Entry Items

Item Name	Definition
ENTRY = PITG	<u>PITG entry</u> : The value field is the file name of the PITG-type TDF.
NAME = <VALUE>	<u>PIT Group name</u> : This parameter names the PIT Group. The entered value must be one to eight alphanumeric characters and unique across all PITG names.
CYCLE = <VALUE>	<u>Cycle time</u> : This value is the time increment for scheduling (processing frequency) of this PITG. Value entered must be an integer, -1 through 4096, and an optional time suffix code that defines the unit of time. The default for the optional time suffix code is "S." Valid suffix codes are: S - Seconds M - Minutes H - Hours Examples: <ul style="list-style-type: none">• For a time duration of 12 minutes, enter 12M.• For 30 seconds, enter 30S, or 30.• For Exclusive scanning (fast as possible, with no other PITs on the port being updated while in this mode), enter 0.• To not schedule, enter -1.
CYC_ALM = <VALUE>	<u>Alarm count</u> : This value is the number of cycles behind that the CLM scheduler would have to be before a warning is journalled. Value entered must be an Integer 0 through 256. Default is 0, which inhibits this alarm message capability.
END	<u>End of entry</u> : This defines the end of the input for one PITG entry.

5.7 Point Interface Table Build Entry

PIT build entry table Table 5-4 defines the user input to create a Point Interface Table (PIT) interface structure.

Table 5-4 PIT Build Entry Items

Item Name	Definition
ENTRY = PIT	<u>PIT Entry</u> : The value field is the file name of the PIT-type TDF.
NAME = <VALUE>	<u>PIT name</u> : This parameter names the PIT. The entered value must be one to eight alphanumeric characters and unique across all PIT names.
CYCLE = <VALUE>	<u>Cycle time</u> : This value is the time increment for scheduling (processing frequency) of this PIT. Value entered must be an integer, -1 through 4096, and an <i>optional</i> time suffix code that defines the unit of time. The default for the optional time suffix code is "S." Valid suffix codes are: S - Seconds M - Minutes H - Hours Examples: <ul style="list-style-type: none">• For a time duration of 12 minutes, enter 12M.• For a time duration of 30 seconds, enter 30S.• To not schedule, enter -1.• For Exclusive scanning (fast as possible, with no other PITs on the port being updated while in this mode), enter 0.• To schedule exclusively, enter 0.
CYC_ALM = <VALUE>	<u>Alarm count</u> : This entry is the alarm limit: the number of cycles behind that the scheduler will have to be before a warning is issued. Value entered must be an integer 0 through 256. Default is 0, which inhibits this alarm message capability.
SLO_DWN = <VALUE>	<u>Slow process rate</u> : This Integer value defines the number of normal data collection cycles to process before processing parameters defined as slow parameters. This permits processing a select set of parameters (slow parameters) at a slower rate than other parameters in the same PIT, in order to conserve processing time and communication bandwidth. Default is 0, which inhibits the slow processing capability for this PIT. (Also refer to the SLO parameter, both input and output functions, within the PITE template).
NDT_NAME = <VALUE>	<u>NDT name</u> : This entry is the name of the NDT related to this PIT. The NDT must have been previously defined.
GRP_NAME = <VALUE>	<u>PITG name</u> : This entry is the name of the associated PITG. This is an optional entry. If entered, this PIT must be a member of the named PITG. If the PIT is not a member of a PITG, leave blank.
END	<u>End of entry</u> : This defines the end of the input for one PIT entry.

5.8 Point Interface Table Entry Build Entry

PITE build entry table Table 5-5 defines the user input to create a PITE interface structure.

Table 5-5 PITE Build Entry Items

Item Name	Definition
ENTRY = <VALUE>	<u>PITE entry</u> : The value field is the file name of a PITE-type TDF.
NAME = <VALUE>	<u>Point name</u> : This entry names the CLM image point that is associated with this entry. Mandatory entry; Value entered must be 1 to 16 alphanumeric characters. Note : The length of the image point name may depend on the LCN release you are using.
PIT_NAME = <VALUE>	<u>PIT name</u> : This entry is the name of the associated PIT. The PITE being defined must be a member of the named PIT. Mandatory entry; Value entered must be one to eight alphanumeric characters, and equal to a previously defined PIT name.
"xxxx" = <VALUE>	<u>Application specific parameter</u> : The "xxxx" symbol represents a one to eight character value that corresponds to an equivalent template definition in the TDF. Use of variable parameters in a specific application's PITE entry type is dependent upon the non-TPN network being interfaced to the CLM. When additional user input is required for a specific PITE entry type, this parameter is mandatory. User input is defined in the corresponding TDF definition.
END	<u>End of entry</u> : This defines the end of the input for one PITE entry.

5.9 Builder Input File Examples and Analysis

Introduction

This subsection has two sets of examples of Builder Type Files:

- Source statement entries in a BIF
- CLMI board and port TDFs

These two sets of examples are followed by a brief analysis.

Source statement entry examples

The following is an example of source statement entries in the BIF. Please note the syntax/format guidelines included in the comments section below.

```
*****
*      ----- (EXAMPLE) -----
*      Builder Input File - Project ABC
*      (Example source input for the CLM Configuration Builder)
*      (End of header comments section)
*****
ENTRY   = CLMI           **Used for CLMI, CCP, and UCIM
NAME    = CLMI_1
A1_CLMI = 1
FAILOVER = N
END
*
ENTRY   = PORT
NAME    = PORT_1
A1_CLMI = 1
A2_PORT = 1
FAILOVER = N
*** Note: The following 6 additional parameters are also
*** required for a Port NDT Template definition;
*** refer to Network Definition Table (NDT)
*** Template below for detailed information
BAUD    = 4
DBITS   = 8
SBITS   = 1
PARITY  = 0
RETRY   = 2
TIMEOUT = 5
END
*
ENTRY   = DEV
NAM     = DEV_1
A1_CLMI = 1
A2_PORT = 1
A3_DEV  = 5
END
*
*** Note: Multiple source statements can be on one line:
ENTRY = PITG; NAME = 5SEC; CYCLE = 5S; CYC_ALM = 3; END
*
ENTRY   = PIT
NAME    = PIT_5SEC
CYCLE   = 5S
CYC_ALM = 3
SLO_DWN = 180
NDT_NAME = DEV_1
GRP_NAME = 5SEC
END
*
```

Continued on next page

5.9 Builder Input File Examples and Analysis, Continued

Source statement entry examples, continued

```
ENTRY   = PID_PT
NAME    = PT345
PIT_NAME = PIT_5SEC
SIG_ADDR = 1056
END
* ***** (End of source BIF example) *****
```

The following is an example of source statements in a Builder Input File used to delete the NDT entries on a single device network. Note the bottom-up order of deletion.

```
*      ----- (EXAMPLE) -----
*      BUILDER INPUT FILE - DELETE
*
*      DELETE = DEVICE
*      NAME = DEV_1
*      END
*
*      DELETE = PORT
*      NAME = PORT_1
*      END
*
*      DELETE = CLMI
*      NAME = CLMI_1
*      END
*
```

CLMI board and port TDF examples

Corresponding template files for the first two entries in the previous source BIF example are given below. These are included here to convey the relationship between the BIF and the corresponding template file that the CLM Configuration Builder uses to validate BIF source.

For a complete description of the TDF, refer to Section 6, Interface Structure Templates and Template Definition Files.

```
*****
*      ----- (PARTIAL EXAMPLE) -----
*      Template Definition File - Project ABC
*      (Example templates for the first two entries in
*      the previous Builder Input File source example)
*      (End of header comments section)
*****
*
*      TDF TEMPLATE - NDT-TYPE FOR CLM INTERFACE BOARD
*      NETWORK COMPONENT
ENTRY   = A8/CLMI
TBL_TYPE : P/NDT
ENT_SIZ : I/
NAME    = A8/
```

Continued on next page

5.9 Builder Input File Examples and Analysis, Continued

CLMI board and port TDF examples, continued

```
ENT_NUM : I/
NUM_KEYS : I/0
A1_CLMI=I(1-2)/1
A2 : I/-1
A3 : I/-1
A4 : I/-1
A5 : I/-1
A6 : I/-1
A7 : I/-1
A8 : I/-1
OPT   : I[ UPLOAD:L1/N; DOWNLOAD:L2/N; RBE:L3/N; FAILOVER=L4/N;
        AUTOSYNC:L5/N; ALT_DISP:L6/N; STARTSEC:L7/N ]
UP_STS : I/;  DWN_STS : I/
FAIL_TM : T/-1;  FAIL_AT:I2/
STS     : I/3
DET_STS : I30/
STA     : I/3
END
*
*   TDF TEMPLATE - NDT-TYPE FOR INTERFACE BOARD PORT
*       NETWORK COMPONENT
ENTRY   = A8/PORT
TBL_TYPE : P/NDT
ENT_SIZ : I/
NAME     = A8/
ENT_NUM : I/
NUM_KEYS : I/4
A1_CLMI=I(1-2)/1; A2_PORT=I(1-4)/1
A3 : I/-1
A4 : I/-1
A5 : I/-1
A6 : I/-1
A7 : I/-1
A8 : I/-1
OPT     : I[ UPLOAD:L1/N; DOWNLOAD:L2/N; RBE:L3/N;
        FAILOVER = L4/N ]
UP_STS : I/;  DWN_STS : I/
FAIL_TM : T/-1;  FAIL_AT:I2/
STS     : I/3
DET_STS : I30/
STA     : I/3
* *** The following are 6 additional parameters required
* *** for a Port NDT *
*
```

Continued on next page

5.9 Builder Input File Examples and Analysis, Continued

CLMI board and port TDF examples, continued

```
KEY1_SZ : I/1
BAUD    = I(0-6)/4
KEY2_SZ : I/1;    DBITS  = I(7-8)/8
KEY3_SZ : I/1;    SBITS  = I(1-3)/1
KEY4_SZ : I/1;    PARITY = I(0-2)/0
* (PARITY VALUE : NONE=0,ODD=1,EVEN=2)
KEY5_SZ : I/1;    RETRY  = I/2
KEY6_SZ : I/1;    TIMEOUT = I/5
*
END
* ***** End of partial TDF example *****
```

Analysis – Builder Input File

The following is a brief analysis of the first two entries in the BIF source example and their corresponding template definitions:

- Templates for each type of CLM interface structure definition must be completed and available to the CLM Configuration Builder in the configured media path prior to commencement of the interface structure build process.
 - For each input, the first BIF source input statement (ENTRY = ...) must match the first statement in a corresponding TDF. The entry field value name also serves as the file name of the TDF.
 - Every BIF source statement begins with a 1-8 character name that must have a match in the corresponding template.
 - There are more definitions in the corresponding TDF than there are in the total BIF entry. Only the definitions that contain an equal sign (ITEM=...) in the template require a user source statement in the BIF.
 - The sequence of source statements in the BIF must be in the same sequence as the corresponding TDF.
 - Definitions in the template that contain a colon ":" are used by the Builder to complete the interface structure definition and allocate it to Network Memory.
 - Multiple statements on the same line must be separated by a semicolon (;).
 - All characters after an asterisk (*) are treated as commentary by the Builder.
 - The END statement must be the last source statement for a BIF entry definition set.
-

5.10 CLM Configuration Builder Displays

Display screen layout Figure 5-3 shows the layout for various Builder display screens. Each Builder display maintains the same screen layout, segmented into 12 areas, as numbered below.

Figure 5-3 Configuration Builder Display Layout

The diagram illustrates the layout of the Configuration Builder display, segmented into 12 numbered areas. The layout is as follows:

- Area #12: . . . (# 1 2) . . .
- Area #1: CLM CONFIGURATION BUILDER
- Area #2: > Select Builder Option
- Area #3: ENTRY TYPE
- Area #4: (Empty box)
- Area #5: (Empty box)
- Area #6: FILE INPUT
- Area #7: BACKBUILD
- Area #8: (Empty box)
- Area #9: (Empty box)
- Area #10: ENTRY DATA
- Area #11: BUILD DATA
- Area #12: (Empty box)

Additional text within the layout includes:

- CLM ID: 55
- ENTRY TYPE:
- DET STATUS
- STATUS : OK

6439

Upper screen layout Area #1 is a Builder system-generated message line that normally contains prompts, in-process status messages, and completion messages. In addition, this line also includes error conditions and termination notices, as required.

The area #1 message line is closely coordinated with area #2, where the Builder system displays a variable length box for user input when required, and function targets numbered #3 through #9.

When specific control information (i.e., file name, entry type, entry name, etc.) is requested, message line #1 begins with "Enter...", and the appropriate length box displays in area #2. The cursor positions here, awaiting specific input data. Additional information on the specific input data may be available in area 12.

Continued on next page

5.10 CLM Configuration Builder Displays, Continued

Upper screen, continued

When the area #1 message line begins with "Select...", the Builder is requesting selection of a specific function target from those available on the current display. Functional targets are indicated by descriptive text inside the target box and appear when the function represents an available option on the current display.

For example, the area #1 message line states "Select Builder Option" while function targets #3, #6, and #7 display descriptive text. This denotes that these three options are now available. When a function target is selected, the next screen update changes the selected target's background color to white, signifying the function is in progress.

Other function targets

Additional function targets highlighted in green displays, as required, to depict available options on the current display. For example, function targets labeled Build or Delete may be highlighted in green if these options are available at the current display level. Function targets shown on the previous display are removed from the current display if they no longer apply.

During the normal build process, function targets are selected and respond to the Builder's specific input data requests for control information until the Builder reaches a detailed display level. At this point, an existing entry (i.e., specific PIT, PITE, etc.) is available for review or edit; data for a new entry can also be entered.

Lower screen

Area #10, labeled Entry Data in Figure 5-3, represents the location on the Builder displays for edit or input of detailed data associated with an entry.

Area #11, labeled Build Data, represents specific error messages, data, and formatting information that correspond to entry data lines in area #10. For example, if a syntax error in parameter data for a PITE entry (area #10) is detected by the Builder, the error message displays on the same line in area #11.

Between area #10 and area #11, a column labeled CLR exists that contains a clear target for each data line for an entry. Selection of a clear target for any line clears or blanks out the value to the right of the equal sign under Entry Data (area #10) on the line selected.

Area #12, located just above the Builder displays represents one additional system-generated message line that is optionally used by the Builder to provide additional guidance. When used, the Builder typically displays the size range for specific input data requested by the "Enter..." statement on message line #1.

5.11 Builder Function Targets

Nine Builder target options

There are nine Builder options available. These are displayed as targets and are numbered areas #3 through #9 in Figure 5-3. The function of the target is displayed within the target's box. The following is a description of each function target in the order of appearance on the Builder displays:

WARNING

The Builder display is an interactive functional process that relies on specific cursor positioning to successfully interact with the BUILD task. The cursor position should NOT be altered either by touch screen OR keyboard actions while the BUSY status is present on the display.

ENTRY TYPE
(#3)

Selection of the **ENTRY TYPE** target informs the Builder that you desire to work in the interactive or single entry Builder mode. The system responds with a request to enter the Entry Type in the data input box located in area #2. The Builder then matches your input with a TDF name from the configured path name.

After locating the template file, the Builder displays the template fields that require user input in area #10. If the template is not found, an appropriate error message is generated. The HELP button can be used to gain information about the template files available. Successful execution of this step invokes the **ENTRY NAME** target

ENTRY NAME
(#4)

Selection of the **ENTRY NAME** target prompts the Builder for input of the name of the entry. The system responds with a request to enter the Entry Name. The input represents the name assignment for this entry. The system places the input in area #10 at the name definition.

If the entry is a PITE, the **PIT NAME** target is invoked. Otherwise, if the named entry exists, the actual entry data is retrieved and displayed in area #10. If the entry does not currently exist, default values from the template file are displayed in area #10. Syntax errors are reported. Help is available through the HELP button.

Continued on next page

5.11 Builder Function Targets, Continued

Note: Selecting the **ENTRY NAME** target after a successful retrieval or BUILD leaves the Build data unchanged. That is, data input for the last entry are the defaults. To bring up the original TDF defaults, select from the **ENTRY TYPE** target for the next build. This is useful when building multiple points of the same entry type.

Nine Builder target options, continued

PIT NAME
(#5)

Selection of the **PIT NAME** target prompts the Builder for input of a PIT name for the associated entry. The actual PITE is displayed if it exists, otherwise the default values of the template file are displayed in area #10. Help is available through the HELP button.

Note: Selecting the **PIT NAME** target after a successful retrieval or BUILD leaves the Build data unchanged. That is, data input for the last entry are the defaults. To bring up the original TDF defaults, select from the **ENTRY TYPE** target for Next Build. This is useful when building multiple points of the same entry type.

BREAK
(#5)

Selection of the **BREAK** target requests the Builder to interrupt the current build process, when the Builder is processing either a **FILE INPUT** or **BACKBUILD** function. Select this option carefully, because the Builder aborts the current process and returns to the initial Builder display. Note that the Builder attempts to complete the entry it is currently processing before initiating the Break action.

FILE INPUT
(#6)

Selection of the **FILE INPUT** target prompts the Builder for input of a file to the build process. The system responds with a request to enter the file pathname and then searches for the file on the specified medium. If the file is located, the Builder displays a request to select the **BUILD** target to start the process. If the file is not located or accessed, an appropriate error is generated. The DET STATUS item gives the error-specific reason. The HELP button can be selected to gain information on the error cause. Refer to Builder HELP.

Note: During the file input build process, the entry name of the current entry being processed is displayed in area #10.

Continued on next page

5.11 Builder Function Targets, Continued

Nine Builder target options, continued

BACKBUILD
(#7)

Selection of the **BACKBUILD** target requests the Builder to perform a backbuild process. The BACKBUILD option generates a BIF using the current CLM interface structures in Network Memory as input definitions to the process. The system asks you to enter a file pathname to which the Builder writes the results.

If the file name entered already exists, the Builder warns the user that the file exists and the process writes backbuild results over the existing contents. If the file name is new, the Builder creates the new file and writes the backbuild results in it. After the system either locates the requested file or creates a new one, the Builder displays a request to select **BUILD** to start the backbuild process. The DET STATUS on the Builder display increments as each successful entry is processed.

Note: During the actual Backbuild process, the **BREAK** target is highlighted on the CLM Configuration Builder display. You can BREAK the Backbuild process at any time, which requests the Builder to stop at the completion of the current interface structure and close the output file. The output file contains a comment noting the demand break. Refer to the **BREAK** target description. Help is available through the HELP button. Refer to Builder HELP.

DELETE
(#8)

Selection of the **DELETE** target is available only at the Detailed display level and applies to the current entry on the Detail display. The Builder deletes the interface structure from Network Memory.

The Builder retains the data for the deleted item. The item can be rebuilt by reselecting either the **PIT NAME** or the **ENTRY NAME** targets.

Continued on next page

5.11 Builder Function Targets, Continued

Nine Builder target options, continued

SKIP
(#8)

Selection of the **SKIP** target requests the Builder to bypass the current entry on the Detail display, and move to the next entry from the input file being used in the file input build process. This option is available when the Builder has detected an error in the current entry from an input file. The alternative is to edit the current entry and select the **BUILD** target to validate the revised entry.

BUILD
(#9)

Selection of the **BUILD** target requests the Builder to accept the current entry on the Detail display, and if valid, store the appropriate CLM interface structure in Network Memory. If the Builder detects an error in the current entry, an appropriate error message is displayed under Error Data (area #11) on the same line where the error is located.

The **BUILD** option is available in the on-line mode, and also in the file input mode, after the Builder detects an error in an entry within the BIF input file being processed.

Note: The **BUILD** target is also used to initiate the actual file input and backbuild processes, after one of these latter two targets has been selected and the appropriate file made current. (Refer to the description under these targets for details.)

Continued on next page

5.11 Builder Function Targets, Continued

Builder Button Actions and Definitions

HELP	This button provides a description of the particular step, phase, or error being processed. Select again to clear.
PAGE FWD	Allows the user to access the next set of user visible parameter fields for view and/or modification.
PAGE BACK	Allows the user to access the previous set of user visible parameter fields for view and/or modification.
DISP BACK	Selecting once reinitializes the builder process. This means close all open files and reset the Builder screen to the initial selections. Selecting a second consecutive time exits the builder process and returns to the CLM Status display.
DISP FWD	Allows exit of the builder process to the CLM Configuration page.

5.12 Builder Display Messages and Prompts

Messages and prompts

Near the top of various displays, the Builder outputs a variety of messages indicating the system is processing, self-explanatory warnings, or prompts requesting user interaction.

The following list is presented merely to familiarize you with these messages and prompts before you encounter them.

BREAK detected. Operation Terminating...
Enter associated PIT Name.
Enter Entry Name.
Enter Entry Type.
Enter File Pathname. Default: NET> >FILENAME.ZI
Entry Exists. Select DELETE or modify data and select BUILD.
Entry not found. Edit data and select BUILD to ADD new entry.
Error! Correct and select BUILD to retry.
Error! Correct and select BUILD to revalidate or select SKIP
Error! Select SKIP to continue or BREAK to abort.
Error! Select SKIP to continue or BREAK to abort.
Error! Select SKIP to leave existing entry or BUILD to replace entry
No entries were processed! Select Builder option.
Nonrecoverable error! Process aborted.
Nonrecoverable error encountered. Backbuild Aborted!
Nonrecoverable Network Memory Error. Delete not done.
Operation Complete. Select option.
Operation DEMAND aborted. Select option.
Processing...Do not touch screen (except BREAK).
Processing...Do not touch screen.
Select BUILD to start Backbuild processing.
Select BUILD to start processing.
Select Builder option.
Select Entry Name.
Select PIT Name.
Unable to Delete. Subordinate entries exist!
Unable to process entered Entry Name!
Unable to process entered Entry Type!
Unable to process entered file name.
Unable to process entered file Pathname.
WARNING: Named file exists! Select BUILD to start Backbuild processing.

5.13 Builder Status Messages

Alphabetical listing

Builder displays output various status and error messages during user interaction and processing. Table 5-6 has a complete list of messages.

Table 5-6 Alphabetical Listing of Builder Status Messages

Message	Explanation
BIF STRUCTURE ERROR	The Builder process has encountered a statement in the input file that does not conform to BIF structural rules. The DET STATUS field is the line number of the BIF statement in error. Refer to BIF general rules.
BUSY BUSY - Checkpoint in Progress BUSY - Entry in Use	The Builder process is executing a requested function. The user should not use the touch screen or keyboard during this time.
DATA ACCESS ERROR	The Builder process has been unsuccessful in attempting to fetch or store to the Builder system database. The DET STATUS field reflects the Data Access Error code. Refer to Data Access Error Codes.
DEFINED ENTRY ALREADY EXISTS	The Builder process has found a matching entry in Network Memory. Building will replace it with the entry in the BIF, while Skipping will leave the current entry.
DUPLICATE ENTRY ENCOUNTERED	The Builder process has not stored the entry into Network Memory because another entry with the same address or primary references already exists.

Continued on next page

5.13 Builder Status Messages, Continued

Alphabetical
listing,
continued

Table 5-6 Alphabetical Listing of Messages (Continued)

Message	Explanation
ENTRY IN USE BY OTHER FUNC	The Builder process is unable to modify or delete the build entry because another system function (such as upload/download) is accessing the existing entry.
ENTRY NAME INVALID	<p>The Builder process is unable to interpret the entered entry name. The entry name does not conform to LCN formats. Entry name formats are:</p> <ul style="list-style-type: none">• Character set is 'A' to 'Z', '0' to '9', and '_'.• At least one non-numeric character is required.• Underscores cannot begin or end the entry name or be consecutive.• Maximum length is eight characters for NDTs, PITS, and PITGS.• Maximum length for pites is 8 (or 16 if 16 character tagname option on LCN.)
ENTRY TYPE/NAME MISMATCH	The Builder process has found the specified entry name, but it is of a different type than that requested.
ENTRY TYPE/TDF MISMATCH	The Builder process has found that the Entry field in the template file does not match the template file name.
FATAL INTERNAL ERROR	The Builder process has encountered a condition that is unexpected and cannot recover from. The function being processed has been terminated.
FILE ACCESS VIOLATION	The Builder process is unable to access the entered file because it is in use by another user or is in a file-protected state.

Continued on next page

5.13 Builder Status Messages, Continued

Alphabetical listing, continued Table 5-6 Alphabetical Listing of Messages (Continued)

Message	Explanation
FILE MANAGER ERROR	The Builder process has encountered a file manager error during file processing. The DET STATUS field reflects the file manager error code. Refer to File Manager Error Codes.
FILE NOT FOUND	The Builder process is unable to find the entered file name on the volume specified.
FILE WRITE ERROR	The Builder process is unable to write to the specified Backbuild file. The DET STATUS field reflects the file manager code. Refer to File Manager Error Codes.
GENERATED ENTRY TOO LARGE	The Builder process is unable to store the entry because the actual size of the entry is larger than the Network Memory page size (4092 words). DET STATUS field has the actual size in words.
ILLEGAL FIELD - TDF	The Builder process has encountered an unexpected invalid field definition in the template file. The DET STATUS field reflects the line number of the statement in error in the template file.
ILLEGAL FILE EXTENSION	The Builder process requires a 'ZI' file extension for File Input and Backbuild files. User has entered a different extension. Note that omitting the file extension will default to the proper one.
ILLEGAL FORMAT - TDF	The Builder process is unable to interpret the format statement for a particular input source line in the template file. The DET STATUS field reflects the line number of the statement in error in the template file.
ILLEGAL PATHNAME ENTERED	The Builder process cannot interpret the entered file pathname. Pathname form is NET>VOL>FILE.ZI or PN:nn>RMrr>Vol>FILE.ZI.

Continued on next page

5.13 Builder Status Messages, Continued

Alphabetical listing, continued Table 5-6 Alphabetical Listing of Messages (Continued)

Message	Explanation
INVALID ENTRY TYPE	The Builder process cannot validate the ENTRY TYPE entered because it does not conform to LCN formats. Refer to ENTRY_NAME_INVALID status message for format rules.
INVALID PRIMARY ENTRY	The Builder process cannot process a primary entry reference on a PIT or PITE because that reference is to a CLMI or PORT device. References must be below the PORT level.
MEMORY ENTRY CORRUPTED	The Builder process is unable to retrieve data from an existing entry in Network Memory. Data retrieved does not conform to expected formats.
MISSING END - TDF	The Builder process has not found the required END field in the template file where it was expected. The DET STATUS field reflects the line number of the statement in error in the template file.
MISSING EXPECTED END - BIF	The Builder process has not encountered an expected END field at the end of a set of entry data. Note that the entry has been built. The DET STATUS field reflects the line number of the statement in error.
NETWORK MEMORY ACCESS ERROR	The Builder process is unable to fetch or retrieve an entry from Network Memory.
NOT ENOUGH MEMORY FOR ENTRY	The Builder process is unable to store the entry into Network Memory because there is not enough AM contiguous memory available.
OK	The Builder process is currently idle. There are no error conditions detected at this time.
OK - NEW ENTRY	The Builder process has been unable to find the specified entry in Network Memory. Processing of the entry data results in the addition of a new entry to Network Memory based on the entered specifics.

Continued on next page

5.13 Builder Status Messages, Continued

Alphabetical listing, continued Table 5-6 Alphabetical Listing of Messages (Continued)

Message	Explanation
OK - FILE ALREADY EXISTS	The Builder process has found that the Backbuild file specified already exists. Continuing with this operation overwrites the file with that name.
OPERATING SYSTEM ERROR	The Builder process has encountered an error in accessing or using the LCN/AM operating system.
PARAMETER ACCESS ERROR	The Builder process has been unable to validate a parameter reference in a PITE. The Data Access error code is reflected in the DET STATUS field. Refer to Data Access Error codes.
PIT NAME INVALID	The Builder process cannot validate the PIT Name entered because it does not conform to LCN formats. Refer to ENTRY NAME_INVALID status message for format rules.
PREMATURE END OF FILE	The Builder process has encountered the physical end of file and data is still required to satisfy an entry action.
PRIMARY ENTRY NOT FOUND	The Builder process is unable to find a reference entry that is a dependency for the current line entry. This could be a higher level device for an NDT, an NDT reference from a PIT or PITE, or a PITG reference from a PIT.
READ ACCESS ERROR - BIF	The Builder process is unable to read the next record in the Input file due to the file manager error reflected in the DET STATUS field. Refer to File Manager Error Codes.
SUBORDINATE ENTRIES EXIST	The Builder process is unable to delete the specified entry from Network Memory because there are entries still existing that reference this entry. These entries must first be deleted.

Continued on next page

5.13 Builder Status Messages, Continued

Alphabetical listing, continued Table 5-6 Alphabetical Listing of Messages (Continued)

Message	Explanation
TDF STRUCTURE ERROR	The Builder process has encountered an error in the format or structure of the template file. The DET STATUS field reflects the line number of the statement in error.
TOO MANY PARAMETERS - TDF	The Builder process has attempted to process a PITE and has processed more parameters than are allowed. Refer to TDF General Rules.
TOO MANY USER-VIS PARS DEF'D	The Builder process has encountered too many user-visible (=) fields during template file processing. Refer to TDF General Rules.
UNABLE TO ACCESS INPUT FILE	The Builder process is unable to access the Builder Input File either because the file does not exist or is in use by another user. Other File Manager errors may attribute to this error, but these are the most common. Refer to File Manager Error Codes.
UNABLE TO ACCESS TDF	The Builder is unable to access the specified template file. The File Manager error code is reflected in the DET STATUS field. Refer to File Manager Error Codes.
UNABLE TO ATTACH PIT TO PITG	The Builder process is unable to assign the PIT to the PITG due to a system error.

Continued on next page

5.13 Builder Status Messages, Continued

Alphabetical listing, continued Table 5-6 Alphabetical Listing of Messages (Continued)

Message	Explanation
UNABLE TO CREATE OUTPUT FILE	The Builder process is unable to create a BACKBUILD file on the HM. The DET STATUS field reflects the File Manager error code. Refer to File Manager Error Codes.
UNABLE TO READ TDF	The Builder process is unable to access the template file for a read operation. The DET STATUS field reflects the File Manager error code. Refer to File Manager Error Codes.
VALIDATION ERROR	<p>The Builder process has detected an error in the entered entry data. The reason for this error is due to one of the following:</p> <ul style="list-style-type: none">• Missing entry - The user did not enter a value in the field and it was a required entry.• Illegal value - The user entered a value in the field that does not match the value type expected.• Value Out of Range - The user entered a value outside of the specified range limits defined on the template file.• Value length too large - The user entered too many characters on an ASCII or Optional parameter definition and exceeded that which was defined in the template file.• Illegal Time Value - The user has entered a value into a time definition field that does not conform to the time format.
VOLUME NOT FOUND	The Builder process cannot find the specified volume on the removable media device or HM.

5.14 File Manager Error Codes

File manager errors

During file processing certain file manager errors may be encountered. These are reflected in the DET STATUS field of the Builder display. Table 5-7 lists the most common errors.

Table 5-7 Common File Manager Error Codes Explained

Error Code	Explanation
01	END of file encountered. Physical end of file.
10	Open files exhausted. Too many files open.
17	Invalid file name. Does not conform to formats.
19	File not found. File is not on specified volume.
21	Access violation. File is in use by other user or is protected.
26	Insufficient storage space. Out of room on volume.
34	File in use. File is being used by another system function.
46	File not open. Operation was attempted before opening file.
52	Invalid Volume Name. Does not conform to volume formats.
53	Volume not found. Specified volume is not on media.
80	Invalid file configuration revision. File is not current revision.

Note: A complete list of file manager error codes is in *Messages Directory*, TDC 1030.

5.15 Data Access Error Codes

Data access errors

During Builder processing, certain data access errors may be encountered. These are reflected in the DET STATUS field of the Builder display. Table 5-8 lists the most common errors.

Table 5-8 Common Data Access Error Codes Explained

Error Code	Explanation
04	Entity name. Name is invalid.
05	Entity ID. Entity does not exist.
14	Insufficient memory. Not enough memory to process.
24	Node unavailable. Data owner node is not on-line.
35	OK with item error. Unable to access the entire list of parameters.
36	Parameter ID error. Parameter does not exist.
39	Point in service. Entity is on-line.
43	Subscript error. Parameter subscript is out of range.
70	Parameter not found. Unable to find specified parameter.
75	Entity name length. Entity name is too long.

Note: A complete list of data access error codes is in *Messages Directory*, TDC 1030.

5.16 Builder Display On-Line HELP

On-Line HELP

The Builder display has access to on-line help information at strategic points in the Builder process. This information is accessed by selecting the <HELP> button on the keyboard.

The Help data is displayed in place of the Builder entry data on the lower half of the display.

Pressing the <HELP> button again returns to the normal display.

Some Help displays have sub-levels available for more detailed data.

Table 5-9 lists the Help references and where they are invoked.

Press the <HELP> button for information on these topics.

Table 5-9 On-line Help Topics

Help Topic	Message Is . . .
BUILDER OPTIONS	SELECT OPTION/BUILD COMPLETE
BUTTON DEFINITION	(FROM BUILDER OPTIONS HELP)
ENTRY TYPE DIRECTORY	(FROM BUILDER OPTIONS HELP)
ENTRY NAME	SELECT ENTRY NAME TARGET
BUILD/DELETE	SELECT BUILD/DELETE
PIT NAME	SELECT PIT NAME
VALIDATION ERROR	CORRECT AND SELECT BUILD TO RETRY
ERROR DEFINITION	(FROM VALIDATION ERROR HELP)
DUPLICATE ENTRY ENCOUNTERED	CORRECT AND SELECT BUILD TO RETRY
PRIMARY ENTRY NOT FOUND	CORRECT AND SELECT BUILD TO RETRY
INVALID PRIMARY ENTRY	CORRECT AND SELECT BUILD TO RETRY
FILE ACCESS ERROR	CORRECT AND SELECT ENTRY TYPE, FILE INPUT OR BACKBUILD TO RETRY

Continued on next page

5.16 Builder Display On-Line HELP, Continued

Table 5-9 On-line Help Topics, continued

FILE ERROR DEFINITION	(FROM FILE ACCESS ERROR HELP)
ENTRY CORRUPTED	CORRECT AND SELECT BUILD TO RETRY
PREMATURE END OF FILE	UNABLE TO PROCESS ENTRY TYPE
ENTRY TYPE/TDF MISMATCH	UNABLE TO PROCESS ENTRY TYPE
ILLEGAL FIELD/TDF STRUCTURE ERROR	UNABLE TO PROCESS ENTRY TYPE
BIF STRUCTURE ERROR	NON-RECOVERABLE ERROR. PROCESS ABORTED.
ENTRY EXISTS	SELECT SKIP TO LEAVE OR BUILD TO REPLACE
FILE EXISTS	NAMED FILE EXISTS
DUPLICATE ENTRY	DUPLICATE POINT NAME OR PARAMETER ADDRESS EXISTS

Section 6 –Interface Structure and TDF

6.1 Defining Interface Structure Templates

Four interface structures

The CLM Configuration Builder requires input definitions to build the following CLM interface structures:

- Network Definition Table (NDT)
- Point Interface Table Group (PITG)
- Point Interface Table (PIT)
- Point Interface Table Entry (PITE)

The Builder validates each structure definition entry submitted, based on a predefined Template Definition File, or TDF.

Section 6 describes the TDF structure.

Note: All CLM Platform and CLM Application packages contain predefined TDFs. These TDFs generally are sufficient to implement and utilize the application. However, there may be instances when the user may want to modify or create a new TDF. This section may be used as a guide for that purpose.

6.2 Template Definition File

Templates in Builder Support directory

The TDF set is unique to each CLM interface type. It contains an individual template file for every CLM interface structure type that is input to the Builder.

Each template is actually a separate file, stored in the directory designated in the builder support pathname of the configuration page.

Types of templates

The application may include multiple templates for each CLM interface.

Each TDF provides the definition of an interface structure to the Builder, including:

- its format;
- constants for the entry type; and
- required user inputs with corresponding validation criteria.

For example, the TDF set includes a template file for each different type of addressable component within the network definition (i.e., board, port, etc.).

If there are eight different levels of addressable components within the total network, the TDF set may contain eight templates, with each template file defining a specific type of addressable component.

The Build process

The Builder uses the first BIF input statement of a definition to identify the template file. The template file is then used to validate the remaining input.

The Builder also uses information in the template to further define the interface structure definition.

When the Build process is completed, the Builder stores the built entry to Network Memory for subsequent CLM reference.

6.3 Template General Entry Format and Rules

Templates are consistent

While the specific TDFs can vary by application, the general entry format and entry definitions remains the same.

Each line in a template file has up to the LCN standard 80 ASCII characters, all of which can be used for template definition entries.

TDF special characters

Within a template source line, the special characters in Table 6-1 are recognized by the Builder.

Table 6-1 TDF Special Characters

Special Character	Definition
*	<u>Comment</u> : The "*" (asterisk) may be at any position on the line. If an asterisk appears in the first column of the line, the entire line is treated as a comment. Within a template line, a comment must start with the asterisk character, signifying the end of processing for that line.
=	<u>Equal sign</u> : This sign indicates to the Builder that there must be an equivalent user input definition for the parameter name. The <VALUE> field follows this sign. (Refer to Template General Format below).
:	<u>Predefined entry</u> : The colon sign indicates to the Builder that the parameter does not require user input. However, the parameter is defined and allocated in Network Memory based on the contents of the <VALUE> field that follows the colon in the template definition.
/	<u>Initialize value</u> : The slash (/) indicates to the Builder that an initial value for this definition follows the sign in the template definition. Note : The slash is required whether a specific default follows it or not.
;	<u>Entry separator</u> : The semicolon is used to separate multiple equivalence entries on the same template line. The semicolon is not required if there is only one equivalence on the line or after the last equivalence.
[]	<u>Multiple value entry</u> : The left and right bracket characters are used to indicate to the Builder that multiple equivalence variables are required to satisfy the definition of one entry. The total enclosure may extend beyond one template line; however, multiple equivalencies on the same line must be separated by a semicolon.

Continued on next page

6.3 Template General Entry Format and Rules, Continued

TDF general format,
continued

Each definition entry in a template file is in the following format:

< ITEM > < = or : > < VALUE >

where:

<ITEM>	Name of the template parameter: Maximum of eight alphanumeric characters allowed.
<= or :>	Either an equal sign "=" or a colon ":" <ul style="list-style-type: none">• Equal sign is used to specify parameters that require user input during the Build process to define the initial value.• Colon is used to allocate parameters that do not require user input during the Build process to complete the definition of the parameter in Network Memory.
<VALUE>	The <VALUE> field represents a compound structure that defines: <ul style="list-style-type: none">• data type;• optional size or optional range; and• initial or default parameter value.

Continued on next page

6.3 Template General Entry Format and Rules, Continued

TDF general format, continued

The three general forms of the VALUE field are:

- Form 1: <TYPE and SIZE>/<INITIAL VALUE>
or
- Form 2: <TYPE and SIZE>/<[STRUCTURE DEFINITION]>
or
- Form 3: <TYPE> <(LIMIT)>/<INITIAL VALUE>

<p><TYPE and SIZE> (Form 1 and Form 2)</p>	<p>Defines the type and size of the data that is to be allocated.</p> <ul style="list-style-type: none">TYPE is a <i>required</i> one-character code, with codes allowed for the data types in the following table:<table border="1" data-bbox="735 705 1411 1089"><thead><tr><th>Code</th><th>Data Type</th></tr></thead><tbody><tr><td>I</td><td>16-bit Integer</td></tr><tr><td>A</td><td>Alphanumeric character</td></tr><tr><td>O</td><td>Optional character</td></tr><tr><td>F</td><td>32-bit Floating point value</td></tr><tr><td>L</td><td>Logical one-bit field (See Note 1)</td></tr><tr><td>T</td><td>16-bit Time duration constant</td></tr><tr><td>P</td><td>Private Builder directive</td></tr><tr><td>S</td><td>Character String (See Note 2)</td></tr></tbody></table><p>Note 1: When the TYPE code = L, an additional numeric suffix (value 1-16) is <i>required</i> to identify the bit in the memory location assigned; e.g., L5 represents the fifth bit.</p><p>Note 2: When the TYPE code = S, an additional nonzero suffix can be applied to designate a mark/use operation. This allows a user-entered string value to be used later in the TDF as needed. The user-entered string value is marked, and then used later with the same mark code on an internal string value. Example: PV = S1/ * marks the value FV : S1/ * uses the marked value</p>SIZE is an <i>optional</i> numeric suffix that defines the number of elements of this data type to be allocated. The SIZE suffix is only permitted with TYPE codes = A, O, or I. When TYPE code = A or O, SIZE must be an even number. For type I the default is 1.	Code	Data Type	I	16-bit Integer	A	Alphanumeric character	O	Optional character	F	32-bit Floating point value	L	Logical one-bit field (See Note 1)	T	16-bit Time duration constant	P	Private Builder directive	S	Character String (See Note 2)
Code	Data Type																		
I	16-bit Integer																		
A	Alphanumeric character																		
O	Optional character																		
F	32-bit Floating point value																		
L	Logical one-bit field (See Note 1)																		
T	16-bit Time duration constant																		
P	Private Builder directive																		
S	Character String (See Note 2)																		
<p>Note: The slash "/" separator is required in all three VALUE forms, if an initial value is defined, after the mandatory TYPE and optional SIZE or (LIMIT) entries.</p>																			

Continued on next page

6.3 Template General Entry Format and Rules, Continued

TDF general format, continued

<p><INITIAL VALUE> (Form 1 and Form 3)</p>	<p>Initial or default value for the field.</p> <ul style="list-style-type: none">• When the TYPE code = I or F, the initial value must be numeric. Default is 0 (zero).• When the TYPE code = I, the value can be prefixed with the character \$ to indicate a hexadecimal value or the character B to indicate an octal value.• When the TYPE code = A or O, the initial value must be a character string, <= than SIZE in length. Default is space(s).• When the TYPE code = L, the initial value must be Y (for yes/on), or N (for no/off). Default is N.• When the TYPE code = T, the initial value represents an integer (1-4096) and an <i>optional</i> time suffix code that defines the unit of time. A value of 0 (zero) specifies exclusive processing, and a value of -1 (minus one) specifies inhibit or never. No default; Valid integer must be entered. The default for the optional suffix code is "S." Valid suffix codes are: S - Seconds M - Minutes H - Hours Examples - When the TYPE code = T, to express a time duration of 12 minutes, define 12M for INITIAL VALUE; 30 seconds would be 30S, or 30 (using default suffix code).• When the TYPE code = P, this value represents a private Builder directive that defines the template type, with one exception. Valid template type initial values are: NDT, PITG, PIT, and PITE. No default; Valid template type must be entered. Note: The one exception for the TYPE code = P, as noted above, is within the PITE template. In the PITE template, the TYPE code = P is used with an initial value that defines the name of the associated PIT structure. In this usage, there is also no default; Valid PIT name must be entered (Refer to PITE Template section).• When the TYPE code = S, the initial value must be a character string. No default; String must be entered. Note: The String field is appended to the CLM interface structure in Network Memory. The String field is preceded by a memory word that contains the number of characters in the character string. The Builder creates a pointer that points to the character string in Network Memory.
<p>Note: The slash "/" separator is required in all three VALUE forms, if an initial value is defined, after the mandatory TYPE and optional SIZE or (LIMIT) entries.</p>	

Continued on next page

6.3 Template General Entry Format and Rules, Continued

TDF general format,
continued

<div><[STRUCTURE DEFINITION]></div> <div>(Form 2)</div>	<p>The right and left brackets "[]" are required, to enclose the structure definition that contains multiple entries having the same general form as the overall template entry form:</p> <div>[<ITEM> <=or:> <TS/V>, <ITEM> <=or:> <TS/V>. . . <ITEM> <=or:> <TS/V>]</div> <p>where:</p> <p>Each <TS/V> entry must have the same form as:</p> <div><TYPE and SIZE>/<INITIAL VALUE></div> <p>Form 1 of the VALUE field above, with the following definition exceptions:</p> <ul style="list-style-type: none">• TYPE codes are limited to: I - 16-bit integer L - Logical one-bit field• The total [STRUCTURE DEFINITION] size is the sum of all explicit SIZE values, plus implicit size defaults, contained within the <TS/V> entries in the [STRUCTURE DEFINITION]. This total size must not exceed the SIZE, as defined for the field itself (within the <TYPE and SIZE> in Form 2 of the VALUE field).
---	---

Continued on next page

6.3 Template General Entry Format and Rules, Continued

TDF general format,
continued

<p><(LIMIT)> (Form 3)</p>	<p>Validation criteria for the field; Available for integer, TYPE = I, data types only. Optional entry; No entry means any integer value allowed. Builder uses integer range, or specific values, defined within parentheses to validate user inputs, per the following forms:</p> <ul style="list-style-type: none">• (LOW-HIGH) where: LOW - Minimum field value HIGH - Maximum field value Note: The dash "-" between the LOW and the HIGH value integers is required in this form.• (INT1, INT2, ...INTn) where: INT1 through INTn are the specific valid field values Note: The commas ", " are optional in this form.• Note: Parentheses are required to enclose validation criteria in both forms.
-------------------------------------	--

Continued on next page

6.3 Template General Entry Format and Rules, Continued

TDF general rules

Rules for entering definition statements in a TDF are summarized below:

- Each set of TDF source statements must begin with the source statement parameter: ENTRY = A8/<VALUE>. <VALUE> is the name of the TDF. (This name will be referenced by the first entry in any set of BIF source statements that use the template).
 - Each TDF name must have a "ZT" extension (i.e., XXXXXX.ZT).
 - All TDFs must reside in the media directory defined in the builder support pathname of the CLM Configuration page.
 - Definitions defined with a colon (:) are not user-visible.
 - Only definitions defined with an equal (=) sign require a user-entered value at structure build time.
 - Sequence of BIF definition statements and TDF user-visible source statements must be the same.
 - A statement cannot extend beyond one line, unless enclosed in brackets.
 - Multiple equivalences within brackets [] can extend beyond one line.
 - Multiple equivalences must be separated by a semicolon (;).
 - When a statement includes equivalences [in brackets], each equivalence must be fully defined on one line.
 - A semicolon (;) delimiter is required between multiple statements on the same line.
 - The maximum number of defined parameters in a PITE is 200.
 - The maximum number of user-visible (=) fields is 50.
 - The END statement must be the last statement in a TDF entry set.
-

6.4 Network Definition Table Template

NDT makeup

The NDT contains a definition of every addressable network component within the CLM realm, including:

- the CLM Interface board(s),
- the ports on each board, and
- each addressable component in the non-TPN network (i.e., gateway, bus, device, subdevice).

In addition, the NDT contains the relationship between these network components, which is a hierarchy from the CLMI board, through its port(s) and through each addressable network component to the lowest level device address.

Each addressable component in the network is a separate entry in the NDT, and each type of component has a corresponding NDT Template Definition File.

When you define a new network component for addition to the NDT, the Builder references the specific template file that corresponds to the component type. The Builder uses this template to validate your input, and provide additional information necessary to build the entry.

Upon completion, the Builder creates the appropriate NDT structure and places it in Network Memory.

The general form and definition of an NDT template file is covered below.

The parameter entries in the file through the “STA” parameter represent parameters that are common to every NDT template file for the CLM Platform. Beyond the “STA” parameter, additional parameters can be added to a specific NDT template to define the characteristics of a specific type of network component. These parameters are called key parameters. Note that the NUM_KEYS parameter must reflect the actual number of these additional parameters.

For example, the common set of parameters suffices to define the NDT template for a board as an addressable component in the network.

Six additional parameters are required to define the NDT template for a port on a board. These additional parameters are defined after the STA parameter, and are referred to as optional key parameters.

Continued on next page

6.4 Network Definition Table Template, Continued

NDT form and definition, continued

Note: In the following general form, only the entries in **bold print** need to be reviewed by the Network Interface Design Engineer during template definition.

The remaining entries should be copied exactly as shown to your template file.

For proper CLM operation, *all templates must include all entries* in the sequence depicted in the template general form.

The following is the general form and definition of an NDT template:

* *** NDT Template General Form ***

*

ENTRY	= A8/
TBL_TYPE	: P/NDT
ENT_SIZ	: I/
NAME	= A8/
ENT_NUM	: I/
NUM_KEYS	: I/
A1	= I/E1
A2	= I/E2
A3	= I/E3
A4	= I/E4
A5	= I/E5
A6	= I/E6
A7	= I/E7
A8	= I/E8
OPT	: I/ [UPLOAD :L1/N; DOWNLOAD :L2/N; RBE :L3/N; FAILOVER :L4/N; ALT_DISP :L6/N; STARTSEC :L7/N]
UP_STS	: I/
DWN_STS	: I/
FAIL_TM	: T/
FAIL_AT	: I2/
STS	: I/3
DET_STS	: I30/
STA	: I/3

*

*** Optional KEY parameters may be added here ***

*** and numbered in the NUM_KEYS field above ***

END

* *** END must be last valid source statement ***

Continued on next page

6.4 Network Definition Table Template, Continued

Table 6-2 lists the NDT Parameter Definitions.

Table 6-2 NDT Parameter Definitions

Parameter Label	Definition
ENTRY = A8/<INITIAL VALUE>	Template name: A separate NDT template file must be defined for each <i>type</i> of network addressable component (i.e., CLMI, PORT, etc.). Value entered must be one to eight alphanumeric characters and unique across all template names in the application. This name also the file name of this template.
TBL_TYPE : P/NDT	COPY this entry; DO NOT modify this entry in your template.
ENT_SIZ : I/	COPY this entry; DO NOT modify this entry in your template.
NAME = A8/	COPY this entry; DO NOT modify this entry in your template. This entry is the name of the device or network component; entered by the user at NDT build time.
ENT_NUM : I/	COPY this entry; DO NOT modify this entry in your template.
NUM_KEYS : I/<INITIAL VALUE>	Number of Key Parameters: Replace <INITIAL VALUE> with an integer that is the number of optional key parameters defined within this NDT template. This value is defined by the engineer when this template is defined (Refer to optional key parameters below).
A1 = I/E1 A2 = I/E2 A3 = I/E3 A4 = I/E4 A5 = I/E5 A6 = I/E6 A7 = I/E7 A8 = I/E8	Address Entry: These eight fields define the Network/Device address for this component. Value entered is the total address, from the top level addressable network component down to this device. For example, if device = first level controller, address entry in the template would be of the form: <ul style="list-style-type: none">• A1_CLMI=I(1-2)/1• A2_PORT=I(1-4)/1• A3_GWY=I(0-16)/0• A4_BUS=I(0-255)/0• A5_CTLR=I(0-127)/0• A6:I/-1• A7:I/-1• A8:I/-1

Continued on next page

6.4 Network Definition Table Template, Continued

NDT parameter definitions, continued

Table 6-2 NDT Parameter Definitions (Continued)

Parameter Label	Definition
	<p>The complete address entry form is: $A1=I/E1; A2=I/E2; \dots A8=I/E8$</p> <p>where:</p> <p>$A_n$ - Address element mnemonic name of the addressable component.</p> <ul style="list-style-type: none">All eight elements are not required for every component; Use only the number of elements required to define the total address.Unused/undefined elements of the address must be entered in the template in the form: $A_n:I/-1$. Therefore, every component's address entry within this template has eight elements, with the relevant elements in the form $A_n=I/E_n$, and the unused elements in the form $A_n:I/-1$. Note: For unused elements, the equal (=) sign must be changed to a full colon (:). <p>I - Identifies the address element as an Integer value.</p> <ul style="list-style-type: none">Desired limits optionally defined within parentheses (), adjacent to the I; Used by the Builder as validation criteria for the element's address value input at build time. <p>E_n - Integer address of the component, for the corresponding address element. For unused elements in the address entry, the value of E_n is -1 (minus one).</p> <p>Note: $A1$ always defines the mnemonic name of the CLMI or CCP Board address element; $E1$ is the CLMI Board number.</p> <p>$A2$ always defines the mnemonic name of the port address element; $E2$ is the port number.</p> <p>Use $A3$ through $A8$ address elements as required to complete the total address for the subject component, with any unused elements taking the form $A_n:I/-1$.</p>

Continued on next page

6.4 Network Definition Table Template, Continued

NDT parameter definitions,
continued

Table 6-2 NDT Parameter Definitions (Continued)

Parameter Label	Definition
OPT : I/ [UPLOAD :L1/N; DOWNLOAD :L2/N; RBE :L3/N; FAILOVER :L4/N; AUTOSYNC :L5/N; ALT_DISP :L6/N; STARTSEC :L7/N; UNUSED :L8/N; UNUSED :L9/N; ALMDAERR :L10/N]	<p>Options Word: These elements define options or attributes of this network component. Replace the N (No/off) with Y (Yes/on) to enable any of the following options:</p> <p>UPLOAD (Y) = This component is UPLine LOADable.</p> <p>DOWNLOAD (Y) = This component is DOWNline LOADable.</p> <p>RBE (Y) = This component supports Report By Exception input.</p> <p>FAILOVER (Y) = In redundant CLM applications, failure of this component on the network initiates a switchover of the CLM to the backup node.</p> <p>AUTOSYNC (Y) = Allows for an attempted restart of the component if in the "hardfail" condition (see FAIL_TM).</p> <p>ALT_DISP (Y) = The Detail Status Display for this component will be the same name as the template used to create this NDT entry.</p> <p>STARTSEC (Y) = In redundant CLM applications, start the interface board on the secondary before the switchover occurs, thus decreasing the time it takes to switchover. This requires that the application running on the interface board to be synchronized with its devices before the failover occurs. Currently, the only application that supports this feature is Modbus Plus.</p> <p>ALMDAERR (Y) = Setting this option bit on any port of either implemented board enables the Data Access PITE error alarming option. Any Data Access level error on input or output of CLM data will result in the status of the associated device change to P_FAIL, if it is not already so. Currently, the only application that supports this feature is Modbus Plus.</p> <p>Note: Refer to the communication and protocol specification for the object network to determine the setting of these parameters.</p>
UP_STS : I/O	COPY this entry; DO NOT modify this entry in your template.
DWN_STS : I/O	COPY this entry; DO NOT modify this entry in your template.

Continued on next page

6.4 Network Definition Table Template, Continued

NDT parameter definitions, continued

Table 6-2 NDT Parameter Definitions (Continued)

Parameter Label	Definition
FAIL_TM : T/<INITIAL VALUE>	<p>Fail Time Duration: This time duration value specifies the time to wait, after the component status (STS) goes to FAIL (soft failure), before setting the component state (STA) to FAIL (hard failure). To continue from a hard failure requires operator action. If the component status returns to OK, before the FAIL_TM time duration has expired, normal operations resume without operator action (unless AUTOSYNC is SET - See previous OPT parameter definition). This time duration is defined by the application engineer when this template is defined.</p> <p>Replace <INITIAL VALUE> with an integer and an optional time suffix code that defines the unit of time. If this value is to be changed on-line, the field must be made user visible by changing the ":" to "=" . The integer value represents the time duration. No default; A valid integer from 1 to 4096 must be entered to define a fail time duration. Two alternate values may be entered: 0 (zero) for immediate fail; or -1 (minus one) for never hard fail the component. The latter two values do not use a suffix code.</p> <p>The default for the optional suffix code is "S". Valid suffix codes are:</p> <ul style="list-style-type: none">S - SecondsM - MinutesH - Hours <p>Example - To express a fail time increment of 10 seconds, define 10S, or 10 (using the default for suffix code).</p>
FAIL_AT : I2/	COPY this entry; DO NOT modify this entry in your template.

Continued on next page

6.4 Network Definition Table Template, Continued

NDT parameter definitions, continued

Table 6-2 NDT Parameter Definitions (Continued)

Parameter Label	Definition
STS : I/3	COPY this entry; DO NOT modify this entry in your template.
DET_STS : I30/	<p>COPY this entry; DO NOT modify this entry in your template.</p> <p>Detailed Status: This represents a 30-Integer array, allocated by the Builder to contain the low level operating status of this network component. The first five elements are reserved for use by the CLM Platform. The remaining integers are reserved for component or device specific use, and may be used to report operational statistics and other status information.</p> <p>Refer to the communication and protocol specification for the object network to determine the definition and use of DET_STS for each component type. Refer also to the CLM Status Definition File section in this manual; This file is designed by the engineer, to allow the CLM Platform to display descriptive text relating to the status values contained in the Detailed Status array.</p>
Note: The NDT structure for a board or port component reserves the first element of the Detail Status array to record the current number of unacknowledged failed devices on the board or port. The second element is reserved to record the number of acknowledged failed devices on the board or port.	

Continued on next page

6.4 Network Definition Table Template, Continued

NDT parameter definitions, continued

Table 6-2 NDT Parameter Definitions (Continued)

Parameter Label	Definition
STA : I/3	COPY this entry; DO NOT modify this entry in your template.
<p>Note: Optional Key Parameters may be added to an NDT Template here</p> <ul style="list-style-type: none">• To provide additional information about a network component.• All NDT Templates have the same structure, up to this point.• Refer to the communication and protocol specification for the object network to determine (any) required Key Parameters and their values.• Key Parameters may or may not be user visible at NDT structure build time.• Key Parameters are defined at template definition.• Each Key parameter is made up of a parameter pair to represent the size of the parameter, and then its value.• The following is an example of a Key Parameter Pair - (Size and Value): <p>KEYn_SZ : I/<INITIAL VALUE></p> <p>This Integer value is the size, in memory words, of this optional key parameter's value, as defined in the next KEYn_VAL parameter entry.</p> <p>KEYn_VAL = <VALUE></p> <p>This value is the actual value of the key parameter. The <VALUE> entered can be any type and range as defined for standard parameters.</p> <p>Note: The NDT Template file for a port on a board requires the following six additional parameters. For your NDT port template file, define your base <i>timeout period</i> for the KEY6 Parameter TIMEOUT. Then, copy these entries; DO NOT modify the first five key parameter entries in your port template file. All six entries are user visible and require user input (if default is not desired) during NDT build time for the port component type.</p>	

Continued on next page

6.4 Network Definition Table Template, Continued

NDT parameter definitions, continued

Table 6-2 NDT Parameter Definitions (Continued)

Parameter Label	Definition
KEY1_SZ : I/1; BAUD = I(0-6)/4	Baud rate: Baud rate for the communication port or channel. Default is 4. Valid values, and their corresponding meanings are: 0 = 300 1 = 1200 2 = 2400 3 = 4800 4 = 9600 (default) 5 = 19200 6 = 38400
KEY2_SZ : I/1; DBITS = I(7-8)/8	Data bits number: Number of data bits in a communication byte. Valid values are 5 to 8. Default is 8.
KEY3_SZ : I/1; SBITS = I(1-3)/1	Stop bits code: Integer value that represents the number of stop bits used to denote the end of a transmitted data byte. Default is 1. Valid values and the corresponding meanings are: <ul style="list-style-type: none">• 1 = 1 stop bit (default)• 2 = 2 stop bits• 3 = 1.5 stop bits

Continued on next page

6.4 Network Definition Table Template, Continued

NDT parameter definitions, continued

Table 6-2 NDT Parameter Definitions (Continued)

Parameter Label	Definition
KEY4_SZ : I/1; PARITY = I(0-2)/0	Data parity code: Integer value that represents the data parity option for transmitted data bytes. Default is 0. Valid values, and their corresponding meanings, are: <ul style="list-style-type: none">• 0 = No parity (default)• 1 = Odd parity• 2 = Even parity
KEY5_SZ : I/1; RETRY = I/2	Retry code: Integer value that defines the number of retries, to attempt on a communication failure, before aborting the request. Default is 2; Default of 2 attempts a total of 3 tries before aborting the request.
KEY6_SZ : I/1; TIMEOUT = I/<INITIAL VALUE>	Timeout period: Integer value that defines the base timeout period for a no-response communication failure before the request is retried. No default; Mandatory entry. Note: All other communication timeouts should be based on this timeout period, to permit adjustment external to the communication package software. Replace <INITIAL VALUE> with an integer that is your base timeout period, in seconds.
This is the end of the optional parameters in the NDT Template file for a PORT.	
END	COPY this entry; DO NOT modify this entry in your template. This source statement defines the end of the current template statement set; Must be the last entry for each template definition.

6.5 Point Interface Table Group Template

PITG usage

A PITG optionally defines a set of PITs from the same non-Honeywell network to be processed as a group.

A PITG represents the highest level entity in the overall point interface structure and, if used, must be defined prior to the definition of member PITs within the group.

A maximum of 50 PITs can be grouped under one PITG, for optimizing data acquisition as required.

There are no network or device-specific parameters in the PITG template. Therefore, one PITG Template Definition File suffices to build any PITG interface structures required.

This allows the template name to be predefined and equal to "PITG," the same as the type of template.

The PITG template definition requires the Network Interface Designer to copy all entries exactly as shown in the general form to your TDF.

The PITG entry may span multiple devices, but not ports.

The following is the general form and definition of the PITG template:

```
* *** PITG Template General Form ***
*
ENTRY      = A8/PITG
TBL_TYPE   : P/PITG
ENT_SIZ    : I/
NAME       = A8/
ENT_NUM    : I/
STS        : I/
CYCL       = T/
TIMER      : I2/
CYC_ROC    : I/
CYC_ALM    = I(0-256)/
GROUP      : I100/
END
* *** END must be last valid source statement ***
```

Continued on next page

6.5 Point Interface Table Group Template, Continued

PITG parameters

Table 6-3 lists the PITG Parameter Definitions.

Table 6-3 PITG Parameter Definitions

Parameter Label	Definition
ENTRY = A8/PITG	COPY this entry; DO NOT modify this entry in your template. PITG is the name of the template and also the file name of the TDF. (The name equals the template-type for the PITG, because there is only one PITG Template Definition File).
TBL_TYPE : P/PITG	COPY this entry; DO NOT modify this entry in your template.
ENT_SIZ : I/	COPY this entry; DO NOT modify this entry in your template.
NAME = A8/	COPY this entry; DO NOT modify this entry in your template. This is the name of the PIT Group defined by this PITG; Entered by the user at PITG build time.
ENT_NUM : I/	COPY this entry; DO NOT modify this entry in your template.
STS : I/	COPY this entry; DO NOT modify this entry in your template.
CYCLE = T/	COPY this entry; DO NOT modify this entry in your template. This entry represents the cyclic rate at which the PITG is to be processed; Entered by the user at PITG build time.
TIMER : I2/	COPY this entry; DO NOT modify this entry in your template.
CYC_PROC : I/	COPY this entry; DO NOT modify this entry in your template.
CYC_ALM = I(0-256)/	COPY this entry; DO NOT modify this entry in your template. This is an alarm notification option, used to force an alarm message if the specific PITG processing falls behind the configured schedule. Entered by the user at PITG build time.
GROUP : I100/	COPY this entry; DO NOT modify this entry in your template.
END	COPY this entry; DO NOT modify this entry in your template. This source statement defines the end of the current template statement set; Must be the last entry for each template definition.

6.6 Point Interface Table Template

PIT explanations

The Point Interface Table (PIT) contains the relationships between CLM image points and their corresponding actual non-TPN network points, along with conversion and acquisition information.

Each point in a PIT is represented by a PITE.

Each PIT defines a set of points grouped together, based on both processing time and physical device address.

A PIT can optionally be a member in one PITG. If a PIT is to be assigned to a PITG, the PITG must be defined prior to the PIT definition.

Grouping network points

The criteria for grouping non-TPN network points into specific PITs is generally based on both:

- the desired processing frequency and
- the physical device address.

Therefore, the total number of points associated with one physical device address can be grouped into multiple Point Interface Tables, with each PIT having its own processing cycle.

If you elect to group non-TPN network points into PITs based on other criteria, multiple PITs can exist per physical device address and/or the same processing frequency.

Note that all points defined within a specific PIT must have the same physical device address and also the same processing cycle.

One PIT template

There are no network or device-specific parameters in the PIT template. Therefore, one PIT Template Definition File suffices to build any PIT interface structures required.

This allows the template name to be predefined and equal to "PIT," the same as the type of template.

The PIT template definition requires the Network Interface Designer to copy all entries exactly as shown in the general form, to your TDF.

Continued on next page

6.6 Point Interface Table Template, Continued

General form and definition

The following is the general form and definition of a PIT template.

General form

```
* *** PIT Template General Form ***
*
ENTRY   = A8/PIT
TBL_TYPE : P/PIT
ENT_SIZ : I/
NAME    = A8/
ENT_NUM : I/
STS     : I/
CYCLE   = T/-1
TIMER   : I2/
CYC_PROC : I/
CYC_ALM = I(1-256)/5
SLO_DWN = I/
SLO_CNTR : I/
NDT_NAME = A8/
NDT_PO   : I2/
GRP_NAME = O8/
PITE_PG  : I/
DA_ALMS  : I/
DEV_ALMS : I/
END
* *** END must be last valid source statement ***
```

Continued on next page

Parameter definitions Table 6-4 lists the PIT Parameter Definitions.

Table 6-4 PIT Parameter Definitions

Parameter Label	Definition
ENTRY = A8/PIT	COPY this entry; DO NOT modify this entry in your template. PIT is the name of the template and also the file name of the TDF. (The name equals the template-type for the PIT, because there is only one PIT template definition file).
TBL_TYPE : P/PIT	COPY this entry; DO NOT modify this entry in your template.
ENT_SIZ : I/	COPY this entry; DO NOT modify this entry in your template.
NAME = A8/	COPY this entry; DO NOT modify this entry in your template. This is the name of the PIT defined by this PIT template; Entered by the user at PIT build time.
ENT_NUM : I/	COPY this entry; DO NOT modify this entry in your template.
STS : I/	COPY this entry; DO NOT modify this entry in your template.

Continued on next page

6.6 Point Interface Table Template, Continued

Parameter definitions,
continued

Table 6-4 PIT Parameter Definitions (Continued)

Parameter Label	Definition
CYCLE = T/-1	COPY this entry; DO NOT modify this entry in your template. This entry represents the cyclic rate at which the PIT is to be processed; Entered by the user at PIT build time.
TIMER : I2/	COPY this entry; DO NOT modify this entry in your template.
CYC_PROC : I/	COPY this entry; DO NOT modify this entry in your template.
CYC_ALM = I(0-256)/	COPY this entry; DO NOT modify this entry in your template. This entry is an alarm notification option, used to force an alarm message if the specific PIT processing falls behind the configured schedule. Entered by the user at PIT build time.
SLO_DWN = I(1-256)/5	COPY this entry; DO NOT modify this entry in your template. Entered by the user at PIT build time. This value defines the number of normal cycles to process, before processing parameters defined as slow parameters. This permits processing a selective set of parameters (slow) at a slower rate than other parameters in the same PIT, in order to conserve processing time and communication band width. Note: Default is 0, which inhibits the slow processing capability for this PIT. (Also refer to the SLO parameter, both input and output functions, within the PITE template).
SLO_CNTR : I/	COPY this entry; DO NOT modify this entry in your template.
NDT_NAME = A8/	COPY this entry; DO NOT modify this entry in your template. This entry is the name of the NDT related to this PIT. Entered by the user at PIT build time. This field can not reference a CLMI or port NDT name.
NDT_PO : I2/	COPY this entry; DO NOT modify this entry in your template.
GRP_NAME = O8/	COPY this entry; DO NOT modify this entry in your template. This is the name of the associated PITG. Optional entry; If entered, this PIT is a member of the named PITG. Entered by the user at PIT build time.
PITE_PG : I/	COPY this entry; DO NOT modify this entry in your template.
DA_ALMS : I/	COPY this entry; DO NOT modify this entry in your template.
DEV_ALMS : I/	COPY this entry; DO NOT modify this entry in your template.

END	COPY this entry; DO NOT modify this entry in your template. This source statement defines the end of the current template statement set; must be the last entry for each template definition.
-----	---

6.7 Point Interface Table Entry Template

Definition

The PITE defines the mapping between a CLM image point and the non-TPN network. The PITE defines the input and output requirements between the CLM image point and the non-TPN network point.

Each PITE within a PIT must have the same processing cycle and the same physical device address.

Each PITE consists of multiple element definitions, as required, to define the characteristics of the non-TPN network point to the CLM.

These elements include:

- the relation between the point's physical device address and its corresponding CLM image point;
- point processing options; and
- a set of parameters that define the point's input and output requirements.

Each non-TPN network point must be defined in a PITE, which must be related to only one PIT.

A separate PITE template file is defined for each type of non-TPN network point (i.e., analog input, analog output, digital input, digital output, etc.).

Since non-TPN network point types are application specific, the input/output parameter definitions included in this CLM Platform document are in general terms.

The PITE template file consists of the following general parameter sets:

- Parameters common to every PITE template
 - Total of 14 parameter definitions
 - Includes through the NUM_PARS parameter
- Image point parameters
 - Each parameter LABEL contains a 2-digit parameter number prefix (P01 through Pnn) to insure a unique parameter label/name
 - There are a total of 16 definitions associated with each input and output image point parameter. Therefore, the first set of 16 contain a "P1_" prefix in the parameter LABEL; the second set of 16 parameter LABELs contain a "P2_" prefix, etc.
 - Each set of 16 parameter definitions is segmented into:
 - 6 general,
 - 5 input operations, and
 - 5 output operation definitions.

(Note PITE template general form)

Continued on next page

6.7 Point Interface Table Entry Template, Continued

PITE parameter sets,
continued

- "KEY" Parameters
 - Optional parameters within a PITE template to provide additional information unique to the application/network point type.
 - Each KEY parameter definition consists of a size and value parameter pair.

ATTENTION

In the following general form, only the entries in **bold print** need to be reviewed by the Network Interface Design Engineer during template file definition. The remaining entries should be copied exactly as shown.

General form

```
* *** *
*
*** (Define one PITE for each image point) ***
*
ENTRY           = A8/
TBL_TYPE        : P/PITE
ENT_SIZ         : I/
NAME            = S/
PT_ID           : I4/
PIT_NAME        = P/
DPOINT          : S/
DPT_AD1         : I/-1
DPT_AD2         : I/-1
NDT_PO          : I/2
RB_TIME         : I/
PT_OPT          : I [ PPS:L1/N ]
NUM_KEYS        : I0
NUM_PARS        : I/
*
*** Define the following for each LCN point parameter ***
*
Pnn_PAR         : S/
Pnn_ID          : I/
Pnn_QUAL        : I2/
Pnn_DPAR: S/
Pnn_AD1         : I/-1
Pnn_AD2         : I/-1
```

Continued on next page

6.7 Point Interface Table Entry Template, Continued

General form,
continued

```
*
*** Define the parameter input operation ***
*
Pnn_ISTA           : I/
Pnn_IFNC           : I [  CYC:L1/N; SLO:L2/N; ECP:L3/N;
                        PPS:L4/N; SYN:L5/N; CN:L6/N;
                        SB:L7/N; RBI:L8/N ]

Pnn_ICNV           : S/
Pnn_IDST           : I/
Pnn_ITVO           : I/
*
* *** Define the parameter output operation ***
*
Pnn_OSTA           : I/
Pnn_OFNC           : I [  CYC:L1/N; SLO:L2/N; ECP:L3/N;
                        PPS:L4/N; SYN:L5/N; RB:L6/N;
                        CBO:L7/N ]

Pnn_OCNV           : S/
Pnn_ODST           : I/
Pnn_OTVO           : I/
*
END
* *** END must be last valid source statement ***
```

Continued on next page

6.7 Point Interface Table Entry Template, Continued

PITE parameter definitions

Table 6-5 PITE Parameter Definitions

Parameter Label	Definition
ENTRY = A8/<INITIAL VALUE>	Template name: A separate PITE template file is defined for each type of AM image point within the application. This name is also the file name of the TDF. Replace <INITIAL VALUE> with 1 to 8 alphanumeric characters that are unique across all template names in the application. It is suggested that the name reflect the image point type, and also include a prefix to avoid naming conflicts when multiple CLMs are used. (An example of an analog input name would be: S1_AI).
TBL_TYPE : P/PITE	COPY this entry; DO NOT modify this entry in your template.
ENT_SIZ : I/	COPY this entry; DO NOT modify this entry in your template.
NAME = S/	COPY this entry; DO NOT modify this entry in your template. This entry names the AM image point that is associated with this entry. Entered by the user at PITE build time.
PT_ID : I4/	COPY this entry; DO NOT modify this entry in your template.
PIT_NAME = P/	COPY this entry; DO NOT modify this entry in your template. This is the name of the associated PIT. The PITE being defined is a member of the named PIT. Entered by the user at PITE build time.
DPOINT : S/	Device point name: DPOINT provides for specifying an ASCII point name in the non-Honeywell device. <ul style="list-style-type: none">• If the non-Honeywell device uses point names, configure this entry as user visible, to be entered by the user at PITE build time as follows: DPOINT = S/• If the non-Honeywell device does not use point names, then configure as NOT user visible by copying the entry to your template as shown: DPOINT : S/

Continued on next page

6.7 Point Interface Table Entry Template, Continued

Parameter definitions, continued

Table 6-5 PITE Parameter Definitions (Continued)

Parameter Label	Definition
DPT_AD1 : I/-1 DPT_AD2 : I/-1	<p>These two fields provide for specifying an integer point address in the non-Honeywell device.</p> <ul style="list-style-type: none"> If the non-Honeywell device uses integer point addresses, configure this entry as user visible, to be entered by the user at PITE build time, as follows: DPT_AD1 = I/0 DPT_AD2 = I/0 If the non-Honeywell device does not use integer point addresses, then configure as NOT user visible by copying the entry to your template as shown: DPT_AD1 : I/-1 DPT_AD2 : I/-1
NDT_PO : I2	COPY this entry; DO NOT modify this entry in your template.
RB_TIME : I/<INITIAL VALUE>	Read back time: Replace <INITIAL VALUE> with the integer value of the number of seconds to wait before performing the read back after output function. This is triggered by output of a parameter configured with the read back option (RB). An input parameter configured with the RB option (RBI) read after RB_TIME seconds. The entry is not user visible. It is defined by the engineer at template definition.
PT_OPT : I/ [PPS:L1/N; CHG_ONLY:L2/N]	<p>Point options entry. This entry is the point options word that contains optional parameters on how the point be processed. Within this 16-bit Integer, each bit may represent a potential option. For the CLM Platform, the following option is defined:</p> <p>PPS:L1/N Replace N with Y to configure the point to process (issued a Point Process Special) at the end of the PITE entry processing. This entry is not user visible; Defined by the engineer at template definition.</p> <p>CHG_ONLY:L2/N Replace N with Y to configure this point to be only updated when a change is detected between the new input data and the value currently residing in the PITE in Network Memory. The default of N allows for the data to be stored to the image point every time, whether it has changed or not.</p>
NUM_KEYS : I/<INITIAL VALUE>	Number of key parameters. Replace <INITIAL VALUE> with an integer that defines the number of optional key parameters defined within this PITE template. This value is defined by the engineer at template definition.

Continued on next page

6.7 Point Interface Table Entry Template, Continued

Parameter definitions, continued

Table 6-5 PITE Parameter Definitions (Continued)

Parameter Label	Definition
NUM_PARS : I/<INITIAL VALUE>	Number of point parameters. Replace <INITIAL VALUE> with an integer that defines the number of point parameters defined within this PITE template. This value is defined by the engineer at template definition. MAX = 200.
Note: The following PITE template definition file segment is repeated for each image point parameter input from or output to the non-Honeywell device. Each image point parameter definition LABEL contains a parameter number prefix (P1 – Pn, where n is the number of image point parameters in the template) to ensure a unique parameter label/name in the template. The following is the general form for an image point parameter:	
Pnn_PAR : S/<INITIAL VALUE>	Replace <INITIAL VALUE> with a string value that names the image point parameter for which this entry is defined. This value is defined by the engineer at template definition.
Pnn_ID : I/	COPY this entry; DO NOT modify this entry in your template.
Pnn_QUAL : I2/	COPY this entry; DO NOT modify this entry in your template.
Pnn_DPAR : S/	Device parameter name. Pnn_DPAR provides for specifying an ASCII parameter name in the non-Honeywell device. <ul style="list-style-type: none"> If the non-Honeywell device uses parameter names, configure this entry as user visible, to be entered by the user at PITE build time as follows: Pnn_DPAR = S/ If the non-Honeywell device does not use parameter names, then configure as NOT user visible by copying the entry to your template as shown: Pnn_DPAR : S/
Pnn_AD2 : I/-1 Pnn_AD2 : I/-1	These two fields provide for specifying an integer parameter address in the non-Honeywell device. <ul style="list-style-type: none"> If the non-Honeywell device uses integer parameter addresses, configure this entry as user visible, to be entered by the user at PITE build time, as follows: Pnn_AD1 = I/0 Pnn_AD2 = I/0 If the non-Honeywell device does not use integer parameter addresses, then configure as NOT user visible by copying the entry to your template as shown: Pnn_AD1 : I/-1 Pnn_AD2 : I/-1

Continued on next page

6.7 Point Interface Table Entry Template, Continued

Parameter definitions, continued

Table 6-5 PITE Parameter Definitions (Continued)

Parameter Label	Definition																								
Pnn_ISTA : I/	COPY this entry; DO NOT modify this entry in your template.																								
Pnn_IFNC : I/ [<table> <tr><td>CYC</td><td>:L1/N;</td></tr> <tr><td>SLO</td><td>:L2/N;</td></tr> <tr><td>ECP</td><td>:L3/N;</td></tr> <tr><td>PPS</td><td>:L4/N;</td></tr> <tr><td>SYN</td><td>:L5/N;</td></tr> <tr><td>CN</td><td>:L6/N;</td></tr> <tr><td>SB</td><td>:L7/N;</td></tr> <tr><td>RBI</td><td>:L8/N;</td></tr> <tr><td>ODC</td><td>:L9/N;</td></tr> <tr><td>OOS</td><td>:L10/N;</td></tr> <tr><td>CC</td><td>:L11/N;</td></tr> <tr><td>DAI</td><td>:L12/N]</td></tr> </table>	CYC	:L1/N;	SLO	:L2/N;	ECP	:L3/N;	PPS	:L4/N;	SYN	:L5/N;	CN	:L6/N;	SB	:L7/N;	RBI	:L8/N;	ODC	:L9/N;	OOS	:L10/N;	CC	:L11/N;	DAI	:L12/N]	<p>This set of parameters defines the basic INPUT function of this parameter entry. This parameter set is defined by the engineer at template definition. A parameter can be defined with the following attributes defining when the parameter is to be input:</p> <p>CYC:L1/N Y = Cyclic*. Replace N with Y to configure the parameter for input at the CYCLE rate.</p> <p>SLO:L2/N Y = Slow cycle*. Replace N with Y to configure the parameter for input at the slow-down cycle rate (As defined by the PIT SLO_DWN parameter).</p> <p><i>*Note: CYC and SLO cannot BOTH be Y.</i></p> <p>ECP:L3/N Y = Exception processing. Replace N with Y to configure the parameter for input from an exception report, either solicited or unsolicited.</p> <p>PPS:L4/N Y = Point PPS processing parameter. Replace N with Y to configure the parameter for input when the point is PPS'd by the operator.</p> <p>SYN:L5/N Y = Database synchronization parameter. Replace N with Y to configure parameter for input on database synchronization (when device is placed in-service).</p> <p>CN:L6/N Y = Change Notification. Replace N with Y to configure the parameter for Change Notification. After database synchronization, this parameter is compared to the last input value in the type_value record and a journal message output if the values are different. In either case, the value is moved to the image point.parameter.</p> <p><i>Note: The SYN attribute must also be set to initialize the type_value record.</i></p> <p>SB:L7/N Y = Set Bad. Replace N with Y to configure the parameter for Set Bad. This sets this input parameter to NaN, if the non-Honeywell device becomes unavailable for input.</p> <p>RBI:L8/N Y = Read Back Input. Replace N with Y to configure the parameter for input after output of a trigger for the read back function. This function waits for the time specified by RB_TIME before performing a demand input of all RBI parameters.</p> <p>ODC:L9/N Y = On Data Change (OPC Client APP only). Replace N with Y to configure the parameter subscription processing when using the OPC Client application with the APP/CLM.</p> <p>OOS:L10/N Y = Out of Service (OPC Client APP only). Replace N with Y to configure the option to set parameters (mapped data items) to BAD (NaN) when the associated device/server is placed out of service.</p> <p>CC:L11/N Y = Continuous Control processing (Spec 200 Emulator APP only). Replace N with Y to allow continuous control interaction between the LCN and the Spec 200 system.</p> <p>DAI:L12/N Y = Inhibit Data Access PITE-level alarming. Replace N with Y to inhibit the optional Data Access PITE-level alarming for this single parameter only.</p>
CYC	:L1/N;																								
SLO	:L2/N;																								
ECP	:L3/N;																								
PPS	:L4/N;																								
SYN	:L5/N;																								
CN	:L6/N;																								
SB	:L7/N;																								
RBI	:L8/N;																								
ODC	:L9/N;																								
OOS	:L10/N;																								
CC	:L11/N;																								
DAI	:L12/N]																								

6.7 Point Interface Table Entry Template, Continued

Parameter definitions, continued

Table 6-5 PITE Parameter Definitions (Continued)

Parameter Label	Definition																		
Pnn_ICNV : S/<INITIAL VALUE>	<p>Replace <INITIAL VALUE> with a coded character string that defines the input conversion code for the parameter. The conversion code specifies the required data transformation between the image point parameter and the non-Honeywell device database. The conversion code must be the same as the Pnn_OCNV parameter, if both are used. The conversion code is entered by the engineer at template definition as a single string of characters in the following format:</p> <p>aaa-bbb</p> <p>where:</p> <ul style="list-style-type: none">"aaa" is the LCN data type"-" is a required string delimiter (dash)"bbb" is the non-Honeywell device data type <p>For the LCN data types, the following conversion codes for "aaa" are defined:</p> <table><tr><th>Code</th><th>LCN data type</th></tr><tr><td>1</td><td>Integer</td></tr><tr><td>2</td><td>IEEE Floating Point</td></tr><tr><td>3</td><td>Logical</td></tr><tr><td>4, Number</td><td>String</td></tr><tr><td>5, Max</td><td>Enumeration</td></tr><tr><td>7, Size</td><td>Floating Point Array</td></tr><tr><td>8, Size</td><td>Logical Array</td></tr><tr><td>9, Max,Size</td><td>Enumeration Array</td></tr></table> <p>"Max" is the maximum value or range "Size" is the number of elements in the array</p> <p>For the non-Honeywell device data type, the conversion code for "bbb" is network device-specific. For a definition of this code refer to either device-specific documentation or the communication and protocol specification for the object network.</p>	Code	LCN data type	1	Integer	2	IEEE Floating Point	3	Logical	4, Number	String	5, Max	Enumeration	7, Size	Floating Point Array	8, Size	Logical Array	9, Max,Size	Enumeration Array
Code	LCN data type																		
1	Integer																		
2	IEEE Floating Point																		
3	Logical																		
4, Number	String																		
5, Max	Enumeration																		
7, Size	Floating Point Array																		
8, Size	Logical Array																		
9, Max,Size	Enumeration Array																		

Continued on next page

6.7 Point Interface Table Entry Template, Continued

Parameter definitions, continued

Table 6-5 PITE Parameter Definitions (Continued)

Parameter Label	Definition
Pnn_IDST : I/	COPY this entry; DO NOT modify this entry in your template.
Pnn_ITVO : I/	COPY this entry; DO NOT modify this entry in your template.
Pnn_OSTA : I/	COPY this entry; DO NOT modify this entry in your template.
Pnn_OFNC : I/ [CYC :L1/N; SLO :L2/N; ECP :L3/N; PPS :L4/N; SYN :L5/N; RB :L6/N; CBO :L7/N]	<p>This set of parameters defines the basic OUTPUT function of this parameter entry. This parameter set is defined by the engineer at template definition. A parameter can be defined with the following attributes, which define <i>when</i> the parameter is to be output:</p> <p>CYC:L1/N Y = Cyclic*. Replace N with Y to configure the parameter for output at the CYCLE rate.</p> <p>SLO:L2/N Y = Slow cycle*. Replace N with Y to configure the parameter for output at the slow-down cycle rate (As defined by the PIT SLO_DWN parameter).</p> <p><i>*Note: CYC and SLO cannot BOTH be Y.</i></p> <p>ECP:L3/N Y = Exception processing. Replace N with Y to configure the parameter for output by way of the store change detect function.</p> <p>PPS:L4/N Y = PPS processing parameter. Replace N with Y to configure the parameter for output when the point is PPS'd by the operator or commanded by a program.</p> <p>SYN:L5/N Y = Database synchronization parameter. Replace N with Y to configure the parameter for output on database synchronization (when the device is placed in-service).</p> <p>RB:L6/N Y = Read Back Trigger. Replace N with Y to configure the parameter as a trigger for the read back function. The read back function waits, for the time specified by RB_TIME, before performing a demand input of all RBI input parameters.</p> <p>CBO:L7/N Y = Check Before Operate. Replace N with Y to configure the parameter for output with the Check Before Operate option (This function is not supported by all non-Honeywell device interfaces).</p>

Continued on next page

6.7 Point Interface Table Entry Template, Continued

Parameter definitions, continued

Table 6-5 PITE Parameter Definitions (Continued)

Parameter Label	Definition																		
Pnn_OCNV : S/<INITIAL VALUE>	<p>Replace <INITIAL VALUE> with a coded character string that defines the output conversion code for the parameter. The conversion code specifies the required data transformation between the image point parameter and the non-Honeywell device database. The conversion code must be the same as the Pnn_ICNV parameter, if both are used. The conversion code is entered by the engineer at template definition as a single string of characters in the following format:</p> <p>aaa-bbb</p> <p>where:</p> <ul style="list-style-type: none">"aaa" is the LCN data type"-" is a required string delimiter (dash)"bbb" is the non-Honeywell device data type <p>For the LCN data types, the following conversion codes for "aaa" are defined:</p> <table><tr><th>Code</th><th>LCN data type</th></tr><tr><td>1</td><td>Integer</td></tr><tr><td>2</td><td>IEEE Floating Point</td></tr><tr><td>3</td><td>Logical</td></tr><tr><td>4,Number</td><td>String</td></tr><tr><td>5, Max</td><td>Enumeration</td></tr><tr><td>7,Size</td><td>Floating Point Array</td></tr><tr><td>8,Size</td><td>Logical Array</td></tr><tr><td>9,Max,Size</td><td>Enumeration Array</td></tr></table> <p>"Max" is the maximum value or range "Size" is the number of elements in the array</p> <p>For the non-Honeywell device data type, the conversion code for "bbb" is network device-specific. For a definition of this code, refer to either device-specific documentation or the communication and protocol specification for the object network.</p>	Code	LCN data type	1	Integer	2	IEEE Floating Point	3	Logical	4,Number	String	5, Max	Enumeration	7,Size	Floating Point Array	8,Size	Logical Array	9,Max,Size	Enumeration Array
Code	LCN data type																		
1	Integer																		
2	IEEE Floating Point																		
3	Logical																		
4,Number	String																		
5, Max	Enumeration																		
7,Size	Floating Point Array																		
8,Size	Logical Array																		
9,Max,Size	Enumeration Array																		

Continued on next page

6.7 Point Interface Table Entry Template, Continued

Parameter definitions, continued

Table 6-5 PITE Parameter Definitions (Continued)

Parameter Label	Definition
Pnn_ODST : I/	COPY this entry; DO NOT modify this entry in your template.
Pnn_OTVO : I/	COPY this entry; DO NOT modify this entry in your template.
Note: Optional Key Parameters can be added to a PITE Template file at this position to provide additional information <ul style="list-style-type: none">All PITE template files have the same structure to this point.Refer to communication and protocol specification for the object network for required Key parameters.Key parameters are defined by the engineer at template file definition.Each Key parameter is made up of a parameter pair representing the size of the parameter and its value.The following is an example of a Key Parameter Pair - (Size and Value):	
KEYn_SZ : I/<INITIAL VALUE>	Replace <INITIAL VALUE> with the size, in memory words, of this optional key parameter's value as defined in the next KEYn_VAL parameter entry.
KEYn_VAL = <VALUE>	Replace <VALUE> with the actual value of the key parameter. The <VALUE> entered can be any type and range as defined for standard parameters. Note: A key parameter may or may not be user visible. If it is not user visible, change the equal sign (=) to a full colon (:) in the above format (KEYn_VAL : <VALUE>).
This is the end of the optional key parameters in the PITE Template	
END	END of template definition: This source statement defines the end of the current template statement set; Must be the last entry for each template definition.

6.8 Template Definition File Examples

This subsection

This subsection includes template definition file examples for all of the CLM interface structures available with the CLM Platform:

- NDT
- PITG
- PIT
- PITE

An example of the corresponding BIF code required for each template type is also included.

The corresponding BIF example follows each TDF example.

NDT template examples

The following are examples of the TDFs required to build the NDT interface structures for a network with the following components:

```
CLMI#1
  PORT#1
    GATEWAY#11
      BUS#21
        DEVICE#31
          SUBDEVICE#00
```

* *** NDT TEMPLATE DEFINITION FILE EXAMPLES ***

*

* CLMI Template file:

*

```
ENTRY          = A8/CLMI
TBL_TYPE       : P/NDT
ENT_SIZ        : I/
NAME           = A8/
ENT_NUM        : I/
NUM_KEYS       : I/0
A1_CLMI        = I(1-2)/1
A2             : I/-1
A3             : I/-1
A4             : I/-1
A5             : I/-1
A6             : I/-1
A7             : I/-1
A8             : I/-1
OPT            : I/[ UPLOAD:L1/N; DOWNLOAD:L2/N; RBE:L3/N;
                  FAILOVER=:L4/N; ALT_DISP:L6/N; STARTSEC:L7/N]
UP_STS        : I/;  DWN_STS : I/
FAIL_TM       : T/1S;  FAIL_AT:I2/
STS           : I/3
DET_STS       : I30/
STA           : I/3
END
```

Continued on next page

6.8 Template Definition File Examples, Continued

NDT template examples, continued

```
* PORT Template file:
*
ENTRY          = A8/PORT
TBL_TYPE       : P/NDT
ENT_SIZ        : I/
NAME           = A8/
ENT_NUM        : I/
NUM_KEYS       : I/4
A1_CLMI        = I(1-2)/1; A2_PORT=I(1-4)/1
A3             : I/-1
A4             : I/-1
A5             : I/-1
A6             : I/-1
A7             : I/-1
A8             : I/-1
OPT            : I[ UPLOAD:L1/N; DOWNLOAD:L2/N; RBE:L3/N;
                FAILOVER=:L4/N ]
UP_STS         : I/;   DWN_STS : I/
FAIL_TM        : T/1S; FAIL_AT:I2/
STS            : I/3
DET_STS        : I30/
STA            : I/3
KEY1_SZ        : I/1
BAUD           = I(0-6)/5
KEY2_SZ        : I/1;   DBITS   = I(7-8)/8
KEY3_SZ        : I/1;   SBITS   = I(1-3)/1
KEY4_SZ        : I/1;   PARITY  = I(0-2)/0
*              (NONE=0,ODD=1,EVEN=2)
KEY5_SZ        : I/1;   RETRY   = I/2
KEY6_SZ        : I/1;   TIMEOUT = I/5
END
*
* GATEWAY Template file:
*
ENTRY          = A8/GATEWAY
TBL_TYPE       : P/NDT
ENT_SIZ        : I/
NAME           = A8/
ENT_NUM        : I/
NUM_KEYS       : I/0
A1_CLMI        = I(1-2)/1
A2_PORT        = I(1-4)/1
A3_GW          = I(0-16)/00
A4             : I/-1
A5             : I/-1
A6             : I/-1
A7             : I/-1
A8             : I/-1
OPT            : I[ UPLOAD:L1/N; DOWNLOAD:L2/N; RBE:L3/N ]
UP_STS         : I/;   DWN_STS : I/
FAIL_TM        : T/1S; FAIL_AT:I2/
STS            : I/3
DET_STS        : I30/
STA            : I/3
END
```

Continued on next page

6.8 Template Definition File Examples, Continued

NDT template examples, continued

```
*
*   BUS Template file:
*
ENTRY      = A8/BUS
TBL_TYPE   : P/NDT
ENT_SIZ    : I/
NAME       = A8/
ENT_NUM    : I/
NUM_KEYS   : I/0
A1_CLMI    =I(1-2)/1
A2_PORT    =I(1-4)/1
A3_GW      =I(0-16)/0
A4_BUS     =I(0-255)/0
A5         : I/-1
A6         : I/-1
A7         : I/-1
A8         : I/-1
OPT        : I/[ UPLOAD:L1/N; DOWNLOAD:L2/N; RBE:L3/N;
              FAILOVER=:L4/N ]
UP_STS     : I/;   DWN_STS : I/
FAIL_TM    : T/1S;  FAIL_AT:I2/
STS        : I/3
DET_STS    : I30/
STA        : I/3
END

*
*   DEVICE Template file:
*
ENTRY      = A8/DEVICE
TBL_TYPE   : P/NDT
ENT_SIZ    : I/
NAME       = A8/
ENT_NUM    : I/
NUM_KEYS   : I/0
A1_CLMI    =I(1-2)/1
A2_PORT    =I(1-4)/1
A3_GW      =I(0-16)/0
A4_BUS     =I(0-255)/0
A5_DEV     =I(1-125)/1
A6         : I/-1
A7         : I/-1
A8         : I/-1
OPT        : I/[ UPLOAD:L1/N; DOWNLOAD:L2/N; RBE:L3/N ]
UP_STS     : I/;   DWN_STS : I/
FAIL_TM    : T/1S;  FAIL_AT:I2/
STS        : I/3
DET_STS    : I30/
STA        : I/3
END
```

Continued on next page

6.8 Template Definition File Examples, Continued

NDT template examples, continued

```
* SUB-DEVICE Template file:
*
ENTRY          = A8/SUB_DEV
TBL_TYPE       : P/NDT
ENT_SIZ        : I/
NAME           = A8/
ENT_NUM        : I/
NUM_KEYS       : I/0
A1_CLMI        =I(1-2)/1
A2_PORT        =I(1-4)/1
A3_GW          =I(0-16)/1
A4_BUS         =I(0-255)/0
A5_DEV         =I(1-125)/1
A6_SDEV        =I(0-255)/0
A7             : I/-1
A8             : I/-1
OPT            : I/[ UPLOAD:L1/N; DOWNLOAD:L2/N; RBE:L3/N ]
UP_STS         : I/;   DWN_STS : I/
FAIL_TM        : T/1S;  FAIL_AT:I2/
STS            : I/3
DET_STS        : I30/
STA            : I/3
END
*
* *** END of NDT Template Definition File Examples ***
```

Continued on next page

6.8 Template Definition File Examples, Continued

NDT template examples, continued

The following is an example of a Builder Input File used to configure the above network in the NDT:

```
*** BIF/NDT BUILD CODE EXAMPLE ***
*
ENTRY      =CLMI
NAME       =CLMI#1
A1_CLMI    =1; FAILOVER = Y
END
*
ENTRY      =PORT
NAME       =PORT#1; A1_CLMI      =1; A2_PORT      =1; FAILOVER = Y
BAUD       =4; DBITS             =8; SBITS        =1
PARITY     =0; RETRY             =2; TIMEOUT      =5
END
*
ENTRY      =GATEWAY
NAME       =GTWY_11
A1_CLMI    =1; A2_PORT          =1; A3_GW          =11
END
*
ENTRY      =BUS
NAME       =BUS_21
A1_CLMI    =1; A2_PORT          =1; A3_GW          =11; A4_BUS      =21
END
*
ENTRY      =DEVICE
NAME       =DEV_31
A1_CLMI    =1; A2_PORT          =1; A3_GW          =11; A4_BUS      =21
A5_DEV     =31
END
*
ENTRY      =SUB_DEV
NAME       =DEV_00
A1_CLMI    =1; A2_PORT          =1; A3_GW          =11; A4_BUS      =21
A5_DEV     =31; A6_SDEV        =00
END
*
*** END of BIF/NDT Builder Input Example ***
```

Continued on next page

6.8 Template Definition File Examples, Continued

PITG template example

The following is an example of the template definition file required to define a new PITG:

```
*** PITG TEMPLATE DEFINITION FILE EXAMPLE ***
*
*   PITG Template file:
ENTRY          = A8/PITG
TBL_TYPE       : P/PITG
ENT_SIZ        : I/
NAME           = A8/
ENT_NUM        : I/
STS            : I/
*              (IDLE=0, PROC=1)
CYCLE          = T/
TIMER          : I2/; CYC_PROC : I/; CYC_ALM = I(0-256)/
GROUP          : I100/0
END
*
*** END of PITG Template Definition File Example ***
```

The following is an example of a Builder input file used to define a new PITG:

```
*** BIF/PITG BUILD CODE EXAMPLE ***
*
ENTRY          = PITG
NAME           = 30SEC
CYCLE          = 30S
CYC_ALM        = 3
END
*
*** END of BIF/PITG Builder Input Example ***
```

Continued on next page

6.8 Template Definition File Examples, Continued

PIT template example

The following is an example of the template definition file required to define a new PIT:

```
* *** PIT TEMPLATE DEFINITION FILE EXAMPLE ***
*
ENTRY   = A8/PIT
TBL_TYPE : P/PIT
ENT_SIZ  : I/
NAME     = A8/
ENT_NUM  : I/
STS      : I/
CYCLE    = T/-1
TIMER    : I2/
CYC_PROC : I/
CYC_ALM  = I(0-256)/
SLO_DWN  = I/
SLO_CNTR : I/
NDT_NAME = A8/
NDT_PO   : I2/
GRP_NAME = O8/
PITE_PG  : I/
DA_ALMS  : I/
DEV_ALMS : I/
*
END
*
* *** END of PIT Template Definition File Example ***
```

The following is an example of a Builder Input File used to define a new PIT table:

```
* *** BIF/PIT BUILD CODE EXAMPLE ***
*
ENTRY   = PIT
NAME     = 5SECSCAN
CYCLE    = 5S
CYC_ALM  = 3
SLO_DWN  = 36
NDT_NAME = RTU_103
GRP_NAME =
*          (Blank = Not a member of a PITG)
END
*
* *** END of BIF/PIT Builder Input Example ***
```

Continued on next page

6.8 Template Definition File Examples, Continued

PITE template example

The following is an example of a PITE Template Definition File.

```
*** PITE TEMPLATE DEFINITION FILE EXAMPLE ***
*
* General PITE Template file:
*
ENTRY   = A8/PID_PT
TBL_TYPE : P/PITE
ENT_SIZ  : I/
NAME     = S/;      PT_ID : I4/
PIT_NAME = P/
DPOINT   : S/
DPT_AD1  : I/-1
DPT_AD2  : I/-1
NDT_PO   : I2/
RB_TIME  : I/
PT_OPT   : I/[ PPS:L1/N ]
NUM_KEYS : I/0
NUM_PARS : I/3
*
P1_PAR   : S/PV;      P1_ID:I/; P1_QUAL:I2/
P1_DPAR  : S/;
P1_AD1   : I/-1
PV_AD2   = I/
*
P1_ISTA  : I/
P1_IFNC  : I/[ CYC:L1/Y; SLO:L2/N; ECP:L3/N; PPS:L4/N
             SYN:L5/Y; CN:L6/N; SB:L7/Y; RBI:L8/N ]
P1_ICNV  : S/2-2; P1_IDST:I/; P1_ITVO:I/
*
P1_OSTA  : I/
P1_OFNC  : I/[ CYC:L1/N; SLO:L2/N; ECP:L3/N; PPS:L4/N
             SYN:L5/N; RB:L6/N; CBO:L7/N ]
P1_OCNV  : S/
***      (Note: Conversion code not req'd) ***
P1_ODST  : I/;      P1_OTVO:I/
*
P2_PAR   : S/SP;      P2_ID:I/; P2_QUAL:I2/
P2_DPAR  : S/;
P2_AD1   : I/-1
SP_AD2   = I/
*
P2_ISTA  : I/
P2_IFNC  : I/[ CYC:L1/Y; SLO:L2/N; ECP:L3/N; PPS:L4/Y
             SYN:L5/Y; CN:L6/N; SB:L7/Y; RBI:L8/N ]
P2_ICNV  : S/2-2; P2_IDST:I/; P2_ITVO:I/
*
P2_OSTA  : I/
P2_OFNC  : I/[ CYC:L1/N; SLO:L2/N; ECP:L3/Y; PPS:L4/N
             SYN:L5/N; RB:L6/N; CBO:L7/N ]
P2_OCNV  : S/2-2; P2_ODST:I/; P2_OTVO:I/
*
```

Continued on next page

6.8 Template Definition File Examples, Continued

PITE template example, continued

```
*
P3_PAR : S/OP;      P3_ID:I/; P3_QUAL:I2/
P3_DPAR : S/;
P3_AD1 : I/-1
OP_AD2 = I/
*
P3_ISTA : I/
P3_IFNC : I/[ CYC:L1/Y; SLO:L2/N; ECP:L3/N; PPS:L4/Y
           SYN:L5/Y; CN:L6/N; SB:L7/Y; RBI:L8/N ]
P3_ICNV : S/2-2; P3_IDST:I/; P3_ITVO:I/
*
P3_OSTA : I/
P3_OFNC : I/[ CYC:L1/N; SLO:L2/N; ECP:L3/Y; PPS:L4/N
           SYN:L5/N; RB:L6/N; CBO:L7/N ]
P3_OCNV : S/2-2; P3_ODST:I/; P3_OTVO:I/
*
P4_PAR : S/OP;      P4_ID:I/; P4_QUAL:I2/
P4_DPAR : S/;
P4_AD1 : I/-1
MODE_AD2 = I/
*
P4_ISTA : I/
P4_IFNC : I/[ CYC:L1/Y; SLO:L2/N; ECP:L3/N; PPS:L4/Y
           SYN:L5/Y; CN:L6/N; SB:L7/Y; RBI:L8/N ]
P4_ICNV : S/5-1; P4_IDST:I/; P4_ITVO:I/
*
P4_OSTA : I/
P4_OFNC : I/[ CYC:L1/N; SLO:L2/N; ECP:L3/Y; PPS:L4/N
           SYN:L5/N; RB:L6/N; CBO:L7/N ]
P4_OCNV : S/5-1; P4_ODST:I/; P4_OTVO:I/
*
END
*
* *** END PITE Template Definition File Example ***
```

BIF input

The following is the user BIF input to define a point for the above example of the PITE Template Definition File:

```
* *** BIF/PITE BUILD CODE EXAMPLE ***
*
ENTRY  = PID_PT
NAME   = PT345
PIT_NAME = PIT_ONE
PV_AD2 = 1032    * These are optional parameters, *
SP_AD2 = 1033    * depending on specific template *
OP_AD2 = 2021    * definition files within the   *
MODE_AD2 = 3324  * application                     *
END
*
* *** END of BIF/PITE Builder Input Example ***
```

Section 7 – Network Interface Design Considerations

7.1 Interface Definition and Users

Interface definition

The Network Interface layer defines the interface between the Process Application layer and the lower layers of the CLM, the platform, and communication protocol layers.

The definition must address the needs of the following three system users.

Three system users

Operator—The Operator is considered the end user of the CLM. The operator interface consists of standard and custom displays reflecting the non-TPN network and its process.

Process Engineer—The process engineer builds image points that reflect the non-TPN network's process data for use at higher levels, for operator monitoring and control.

Network Interface Design Engineer—This engineer defines the non-TPN network to the CLM and the way its data is reflected in the image point database. The needs of this system user are the primary emphasis of this section.

Continued on next page

7.1 Interface Definition and Users, Continued

The Design Engineer's task

The Network Interface Design Engineer needs a good understanding of both TPN and the non-TPN network being interfaced. This engineer should also obtain a working knowledge of the lower layers of the CLM.

In general, the task of the Network Interface Design Engineer is to define the non-TPN network and its process data for use by the Operator and the Process Engineer.

The NDT

A CLM interface data structure called the Network Definition Table (NDT) is used to specify the non-TPN network.

The NDT is similar to a file that contains one record for each addressable component in the subsystem from the interface board and specific port down to the network device. A simple network can consist of only the interface board, a port, and one or more devices. More complex networks can also incorporate other addressable components such as gateways, buses, and subdevices.

In defining the network, the engineer first defines templates that represent the addressable components in the network. Templates are basically forms that are used to specify the types of components in the network.

These templates are called Template Definition Files (TDFs) and are used as input to the CLM Configuration Builder to define entries in the CLM NDT. These entries are then used by the CLM to identify the components and retain status and state information for display and management by the operator.

All NDT entries have a common structure, and most of the data is of common form for display by the CLM Network Status display. However, each NDT entry can also include device specific data that requires a custom translation to display.

This requires that the engineer define and create a translation file called the CLM Status Definition File or STSDF. An STSDF file is created for each network component type. (i.e. PORT.ZS)

AM image points

The network process data is made available to the LCN for display and higher level control through image points in the CLM node.

Note: Though the CLM image point is normally the preferred method of displaying the network process data, other image point types from other AMs or node types can be referenced for input of data. However, all output of data to the network must be initiated from the local CLM image point.

Continued on next page

7.1 Interface Definition and Users, Continued

Image points, continued

Image points can be of any point type conducive to the function of reflecting the network data. Frequently, network data does not map directly into the CLM point structures and requires the addition of Custom Data Segment packages (CDS) to contain the data.

Some CLM image points may also include CL programs to perform special processing on the input or output data before it is made available to the higher level control.

Note that simple image points can be defined without using either CDS or CL programs. The use of CL is a very infrequent requirement.

Image points are connected or interfaced to the network data through a CLM structure called the Point Interface Table Entry (PITE). Like the NDT, PIT Entries are defined using the CLM Configuration Builder and templates.

A PITE defines the mapping between the parameters of one image point and the network data. It defines data conversions, scheduling, and other information used to define the processing of the data.

PIT structure levels

PIT structures are multi-level, with the PIT group or PITG at the top level. The PITG is optional and can be used to define a set of PIT tables that are to be processed as a group. This also allows for group processing of data from multiple devices.

The second level down in the point interface structure is the PIT table. It defines a set of data within one network device that is processed together. Multiple PITs can be defined for one network device.

The PIT Entry or PITE is at the lowest level in the point interface structure. The PITE defines a set of related values from the network device that are to be mapped into one image point.

An image point can have only one PITE entry. However, the PITE entry can include fields that map to multiple image point parameters (i.e., PV, SP, OP and MODE).

7.2 Network Design and Configuration Steps

Non-network TPN

Table 7-1 Design and Configure a CLM Non-TPN Network

Step	Action
1	Obtain and understand the following documents: <ul style="list-style-type: none">• <i>CLM Platform Implementation Manual</i>• CLM communication and protocol specification for the object network• Vendor documentation for the object network
2	Identify all the <i>types</i> of addressable components in the network. <ul style="list-style-type: none">• An addressable network component is an address of a device that the CLM must build into its address structure in order to access the data resident in the end network device.• The <i>types</i> of addressable network components vary with the network. However, all networks have as the first two component types an interface board and a port on the board. In addition, networks have additional addressable components such as Device, Sub-device, RTU, Controller, PLC, Gateway, Bus, etc.• This information is available from the CLM communication and protocol specification for the object network, and vendor documentation for the object network.
3	Define and create a TDF for each <i>type</i> of addressable component in the network. These files are used in building the NDT. <ul style="list-style-type: none">• Detailed information on this step is provided in the Defining Interface Structures section.
4	Define and create a CLM Status Definition File (STSDF) for each addressable component type in the network. This is used by the CLM Network Status display. <ul style="list-style-type: none">• Detailed information on this step is provided in the subsection CLM Status Definition File Design.
5	Transfer the TDF and STSDF files to the HM directory NET>&CUS or the specific pathname defined on the CLM Configuration page for use.
6	Identify each addressable component in the network. <ul style="list-style-type: none">• Each component must be identified for input into the CLM node.
7	Create a BIF with one NDT entry for each of the identified addressable network components. <ul style="list-style-type: none">• The BIF is created using the TPN file editor or other text file editor and then transferred to the HM directory NET>&CUS for use.
8	Build the NDT on the target CLM node. This is performed using the CLM Configuration Builder, which is installed on the CLM node as part of the CLM platform software. <ul style="list-style-type: none">• Instructions for using the CLM Configuration Builder are detailed in the Building CLM Interface Structures section.

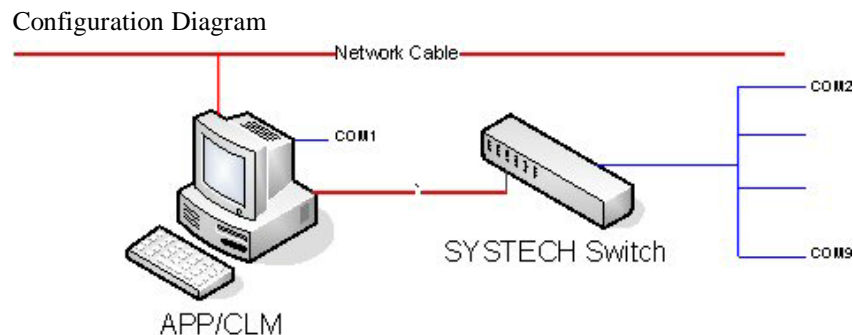
7.3 SYSTECH COM Port Extender

SYSTECH definition	The SYSTECH COM Port Extender is a device, supported on Windows 2003 server and Windows 2008 server (32 bit), and is a replacement for Stallion card. The COM Port Extender supports a maximum of 8 ports.
Reference - External	For Installation and Configuration of the SYSTECH COM Port Extender, refer to the guides you receive with the device.
Note	In Windows 2003, the SYSTECH COM Port Extender needs to be used in place of Stallion Card.

SYSTECH Configuration with Honeywell APP Node

Overview The computer in which the TPS Communication Link Module (CLM) is installed is connected to the existing LAN network. APP server in CLM includes the COM port, communication server, database server, and so on. The SYSTECH switch enables you to extend the COM ports.

Diagram



Configuration Sequence

The following steps describe the network configuration sequence for this scenario.

- The port of the additional network card added can be identified by reading the device name under network connections.
 - The Static IP address assigned to the additional network card should preferably belong to Class C. Ensure that the IP address must not belong to the same subnet of already existing network.
 - The network binding priority of the added network card will be by default, set high, which can be modified through the **Advanced** option under the **Network Connection** window.
- Now, the SYSTECH switch can be used to connect and communicate between multiple COM ports.

7.4 CLM Status Definition File Design

STSDF

The CLM Status Definition File (STSDF) supports the CLM Network Status display to show device specific statistical and status information.

The display is constructed from two parts:

- data from the NDT detailed status array (DET_STS) or Key parameters (KEYn_VAL);
- formatted text from the STSDF. There is one line in the STSDF for each NDT element to be displayed. Refer to the CLM Device Detail Status display.

The detailed status parameter

The data is maintained in the NDT parameters DET_STS, which is defined as a 30-element array of 16-bit integers, and the key parameters KEYn_VAL.

The first five elements of the array are reserved for the CLM Platform. The remaining 25 integer elements can be used to report operational statistics and other status information such as:

- number of bytes transmitted
- number of bytes received
- number of device busy/wait responses
- number of internal SEND busy responses
- total errors this hour or day
- total errors last hour or day

Refer to the communication and protocol specification for the object network for the definition and use of the NDT parameters for each component type.

Each CLM Status Definition File is an ASCII file which is created as part of the Network Interface layer. Its file name must equal the NDT template definition file name that the STSDF relates to, with a file name extension of "ZS" (i.e., CLMI.ZS).

Each STSDF formats the data and adds descriptive text for display on the Component Detailed Status page(s) of the Network Status display.

This feature permits the data in the NDT to be interpreted and displayed as variable length fields from one to five digits in length. This interpretation allows devices with many concurrent state and status conditions to be displayed with descriptive text.

The Component Detailed Status page(s) is displayed with 18 lines per page and 78 ASCII characters per line. Each STSDF contains one format line for each item to be displayed from a specific DET_STS or KEYn_VAL element.

Continued on next page

7.4 CLM Status Definition File Design, Continued

STSDF format, continued

The following is the general form and definition of the STSDF format.

word,start_bit,end_bit,line,column(ccc"text" , ccc"text" ... , "text")

where:

word	The word number in the NDT parameter DET_STS that contains the data to display. The 30 words in DET_STS are numbered 1 through 30 in this format. The 16 bits in each word are numbered in descending order from 15 to 0.
start_bit	The leftmost bit in the data field.
end_bit	The rightmost bit in the data field.
line	The line number in the CDS page(s) to output the data and text.
column	The column number in 'line' to output the data and text.
(ccc"text" , ccc"text"... , "text")	<p>ccc"text" - is an optional conditional display clause, with each clause separated by a comma, and all of the clauses enclosed in parentheses.</p> <p>ccc - is a number that defines the conditions for which the "text" is to be displayed. If the data field matches the number ccc, then the "text" is displayed. ccc should not be defined for the last clause, as the last "text" is displayed if all previous clauses are not selected. Evaluation of the clauses occurs from left to right and stops when a match is found (contents of data field equal ccc).</p> <p>"text" - is a character string that is the description of the field and/or number in the field. The character "#" is used to specify where the number from the data field is to be placed in the text string. Consecutive "#" characters define the width of the data field.</p>

Continued on next page

7.4 CLM Status Definition File Design, Continued

Status Definition File example

The following is an example of a CLM Status Definition File.

```
1,15,0,1,1( 0" # - DETAILED STATUS IS GOOD",
1" # - DETAILED STATUS IS ERROR",
2" # - DETAILED STATUS IS FAILED",
" ##### - UNDEFINED DETAILED STATUS" )
2,15,0,2,1( " ##### - BYTES TRANSMITTED")
3,15,0,3,1( " ##### - BYTES RECEIVED")
4,15,0,4,3( " ### - NUMBER OF DEVICE BUSY/WAIT RESPONSES")
5,15,0,5,3( " ### - NUMBER OF INTERNAL SEND BUSY RESPONSES")
6,15,0,6,1( " TOTAL ERRORS THIS HOUR = ##")
7,15,0,7,1( " TOTAL ERRORS LAST HOUR = ##")
9,15,15,8,01(0"CARD 1 FAILED",1" CARD 1 OK")
9,14,14,8,18(0"CARD 2 FAILED",1" CARD 2 OK")
9,13,13,8,36(0"CARD 3 FAILED",1" CARD 3 OK")
9,12,12,8,54(0"CARD 4 FAILED",1" CARD 4 OK")
9,11,11,9,01(0"CARD 5 FAILED",1" CARD 5 OK")
9,10,10,9,18(0"CARD 6 FAILED",1" CARD 6 OK")
9,09,09,9,36(0"CARD 7 FAILED",1" CARD 7 OK")
9,08,08,9,54(0"CARD 8 FAILED",1" CARD 8 OK")
```

Display example

The following is an example display for the above definition.

```
0 - DETAILED STATUS IS GOOD
21502 - BYTES TRANSMITTED
3031 - BYTES RECEIVED
  4 - NUMBER OF DEVICE BUSY/WAIT RESPONSES
  0 - NUMBER OF INTERNAL SEND BUSY RESPONSES
TOTAL ERRORS THIS HOUR = 6
TOTAL ERRORS LAST HOUR = 1

CARD 1 FAILED  CARD 2 OK   CARD 3 OK   CARD 4 OK
CARD 5 OK     CARD 6 FAILED CARD 7 OK   CARD 8 OK
```

7.5 Image Point Database Design and Configuration Information

Image point design

Note: Image points are created in connection with the specific protocol being implemented and are covered in detail in the user's manual for the specific protocol. Information referencing image points is provided here for continuity of the entire CLM interface process.

Image Points are built in the CLM with parameter connections to one or more signals coming from a non-Honeywell device.

The LCN operator views image points in the same manner that he views points coming from a data acquisition device connected to the LCN. The CLM platform provides the data interface between the LCN and a non-TPN network.

The CLM platform also provides the means to implement and organize these image points into hierarchical PIT structures using build templates.

Where display/console actions are required, the image points provide "specifically defined" input and output to support the network device data point.

Typically, the network device data point types are:

- analog input,
- analog output,
- digital input,
- digital output, or
- counter.

Network data points must be emulated by the standard LCN point types. These points are referred to as image points.

Image points provide faceplates for the display and operation of the points from an associated network controller point.

The values are read by the CLMI software and stored to the point as a function of the PIT processing. Likewise, values are read from the image point and output to the associated network controller point.

Continued on next page

7.6 Image Point Database Design and Configuration Information, Continued

Building Image Points

Table 7-2 Building Image Points in the CLM

Step	Action
1	Obtain and understand the following documents: <ul style="list-style-type: none"> • LCN documentation on AM implementation and the Engineering Personality. • <i>CLM Platform Implementation Manual</i> (this manual) • CLM communication and protocol specification for the object network • Vendor documentation for the object network
2	Perform the Network Design and Configuration. <ul style="list-style-type: none"> • This effort is described in the previous section of this document.
3	Identify all image point <i>types</i> to be used to reflect the non-Honeywell device data points. <ul style="list-style-type: none"> • Typically, non-Honeywell device data points are Analog Input, Analog Output, Digital Input, Digital Output, and Counters, and must be emulated by the standard AM/CLM point types. • These AM points are referred to as image points or CLM image points.
4	Define and create the AM image points, their CDS structures, and CL programs, if any. <ul style="list-style-type: none"> • Additional information on this step is provided in Image Point Operation Overview below.
5	Define and create a TDF for each <i>type</i> of image point. <ul style="list-style-type: none"> • These template files are used in building the PIT structures. • Additional information on this step is provided in Image Point Operation Overview below. • Put the TDF files in the HM directory NET>&CUS.
6	Identify each data point in the network that requires an AM image point. <ul style="list-style-type: none"> • The number and type of data points varies with the network.
7	Build the image points and their point interface structures on the target CLM node. <ul style="list-style-type: none"> • This step is performed using the LCN Data Entity Builder and the CLM Configuration Builder that is installed on the CLM node as part of the CLM platform software. • Instructions for using the LCN Data Entity Builder are detailed in the <i>LCN Data Entity Builder Manual</i>. • Instructions for using the CLM Configuration Builder are detailed in subsection 3.5, Building CLM Interface Structures.

7.6 Image Point Operation Overview

Description

An overview of image point operation is given in the following descriptions.

Image point Build operation

The standard LCN Data Entity Builder is used to define and build the image point.

CLM points can include CDS and CL programs, to augment the standard AM point structure and functionality to reflect the point in the non-Honeywell device.

These image points are usually defined by the CLM Network Interface layer for use by the Process Application layer. The image point definition also includes the TDF file, used by the CLM Configuration Builder to define the input and output connections between the CLM image point and the non-Honeywell device.

After building the image point, the CLM Configuration Builder builds the PITE using the TDF file and user inputs.

After a successful build operation, the point interface is complete and data exchange with the non-Honeywell device should be operative.

Initialization and synchronization

Database initialization or synchronization between the CLM image point and the non-Honeywell device can be configured to occur on the establishment or re-establishment of communications with the device.

The PITE template file defines the point parameters used in the database synchronization process. The point parameters are identified by the entries in the PITE shown in Table 7-3.

Table 7-3 Point Parameter Identification PITE Entry Options

Purpose	PITE Entries
For input/upload initialization	Pnn_IFNC:I/[SYN:L5/Y]
For output/download initialization	Pnn_OFNC:I/[SYN:L5/Y]

Continued on next page

7.6 Image Point Operation Overview, Continued

Initialization and synchronization, continued

This permits parameters to be configured during the synchronization process (when placing the device In Service) for either uploading into the image point or downloading from the image point.

Input parameters can be configured for change notification. This option issues an operator (journal) notification if a database mismatch is detected after synchronization. The input value is always moved to the image point. This option is configured using the following PITE entry:

Pnn_IFNC:I/[CN:L6/Y]

The SET BAD option permits input parameters that can accept the value NaN (Not a Number) to be set if the input value becomes unavailable. This option is configured using the following PITE entry:

Pnn_IFNC:I/[SB:L7/Y]

Cyclic PIT processing

After the database synchronization process is complete, normal PIT cyclic processing starts. This process is directed by schedule information in the Point Interface Table structures (PITs, PITGs and PITEs).

From these tables information may be exchanged, input, or output, on a defined cycle.

A parameter is configured for cyclic processing with one of the PITE entry options in Table 7-4.

Table 7-4 Parameter Configuration PITE Entry Options

Cyclic Processing	PITE Entry Option
Normal input cycle processing	Pnn_IFNC:I/[CYC:L1/Y; SLO:L2/N]
Slow input cycle processing	Pnn_IFNC:I/[CYC:L1/N; SLO:L2/Y]
Normal output cycle processing	Pnn_OFNC:I/[CYC:L1/Y; SLO:L2/N]
Slow output cycle processing	Pnn_OFNC:I/[CYC:L1/N; SLO:L2/Y]

Slow cycle processing, or slowdown, permits parameters to be input or output at a slower rate than the normal cycle.

The slow rate is expressed as a multiple of the normal cycle.

A zero slow rate inhibits the processing of parameters configured for SLO=Y.

A process cycle is configured with both PIT and PITG entries, while the slow down code is only configured in the PIT entry.

The configuration parameters in the PIT are:

CYCLE = T – normal cycle time
SLO_DWN = I – slowdown code

The PITG configuration parameter is:

CYCLE = T – normal cycle time

Continued on next page

7.6 Image Point Operation Overview, Continued

Operator image point interface

Parameters can be configured for output when changed by the operator or the program with the following entry in the PITE:

Pnn_OFNC:I/[ECP:L3/Y]

These parameters must reside on an image point local to the CLM node.

Associated with the operator output change detection processing (ECP) is the read back function (RB). This option forces a demand input of all input parameters configured with RBI=Y in a PITE after the ECP output. The read back function waits for the time specified by RB_TIME before performing the demand input. Read back is configured with the following entry in the PITE:

Pnn_OFNC:I/[ECP:L3/Y; RB:L6/Y]

Also associated with the operator output change detection processing (ECP) is the Check Before Operate function (CBO). This option is protocol specific and not supported by all non-Honeywell controllers. When supported, the communications software issues the request to operate, confirm the receipt of the output request, and then authorize the output operation.

CBO is configured with the following entry in the PITE:

Pnn_OFNC:I/[ECP:L3/Y; CBO:L7/Y]

Demand processing (Process Special)

The User Station operator can demand process an image point by initiating a Point Process Special (PPS) to the image point. All parameters configured with the PPS option are input or output. A parameter is configured for PPS processing with one of the following PITE entry options:

Pnn_IFNC:I/[PPS:L4/Y] - input PPS processing

Pnn_OFNC:I/[PPS:L4/Y] - output PPS processing

PITE template conversion codes

Conversion codes specify the required data transformation between the CLM image point parameter and the non-Honeywell device database. Conversion codes are required for each input and output parameter and defined in the PITE template definition file for the image point. The following PITE parameters are used to define the conversion codes:

Pnn_ICNV:S/ - for input parameters

Pnn_OCNV:S/ - for output parameters

The conversion code is entered as a coded character string consisting of two elements joined with a dash:

- LCN data type and
 - Non-Honeywell device data type.
-

Continued on next page

7.6 Image Point Operation Overview, Continued

PITE template conversion codes, continued

The two conversion codes are entered as a single string in the PITE. The following are the general forms for the PITE entries:

Pnn_ICNV:S/aaa-bbb
Pnn_OCNV:S/aaa-bbb

where:

“aaa” is the LCN data type. For the LCN data types, the following conversion codes are defined:

Code	LCN Data Type
1	Integer
2	IEEE Floating Point
3	Logical
4	String
5,Max	Enumeration
7,Size	Floating Point Array
8,Size	Logical Array
9,Max,Size	Enumeration Array

Max is the maximum value or range.

Size is the number of elements in the array.

“-” is a required string delimiter

“bbb” is the non-Honeywell device data type. The code is network device specific. For a definition of this code refer to the CLM Communication Protocol specification for the non-TPN network.

7.7 PIT Interface Design and Configuration

PIT structure

The Point Interface Table (PIT) structure specifies the connections between the CLM image points and the associated non-TPN network points, along with other conversion and acquisition information. These structures are built and maintained by the CLM Configuration Builder. PITs also serve to group sets of points into tables. The criteria for grouping into tables is based on both processing time and physical device addresses. Multiple groups can be used for both time- and device-based groups.

The highest entity in the PIT structures is the PIT Group or PITG. The PITG optionally links a group of PITs from the same device network or CLMI port that are to be collected as a group. The PIT is a set of points arranged by device that make up a processing group for a device. A device can have several PITs. Each image point in the PIT is one entry or PITE. The PITE consists of one or more parameters that define the interface input and output requirements. The parameter is the lowest element in the PIT structure.

Order of definition

If the PIT is to be assigned to a PIT Group, then the PITG must be defined first. To build a Point Interface Table, the user must first define the PIT and then add PITE(s) to the table.

Using the Builder

Entries are added to the PIT structures through the CLM Configuration Builder.

The Builder requires two inputs to add an entry:

- (1) a TDF, and
- (2) the user inputs, which can be either a BIF or entered through the Builder display.

A TDF serves to define an entry to the Builder. It defines the structure of the table entry, constants for the entry type, and the required user inputs to add an entry to the PIT.

Each PIT component (i.e. PITE) has a separate template file and the name of the file identifies the component.

Section 8 - Image Points

8.1 Standard CLM Point Types and Image Point Selection

Supported types

The CLM supports the AM-based data point types described in Table 8-2.

Table 8-1 CLM-Supported Data Point Types

Data Point Type	Description
CLM Regulatory	<p>The CLM Regulatory point is used to monitor and control analog process variables. Process Variable (PV) and control algorithm functions are supported separately.</p> <p>The CLM Regulatory point is used for analog inputs by configuring for PV data acquisition and null control.</p> <p>Full PV alarming functionality is available.</p> <p>When control processing is enabled with an auto manual or CL algorithm, output processing, including mode control, becomes available.</p> <p>As an image point, the CLM Regulatory point is used to reflect this control function and provide PV alarming at the LCN.</p>
CLM Numeric	<p>The CLM Numeric point provides a simple interface for numeric data.</p> <p>As an image point, it can be used for input and/or output of a single numeric floating point value.</p>
CLM Flag	<p>The CLM Flag point provides a simple interface for a two-state discrete variable.</p> <p>As an image point, it can be used for input and alarm of a two-state discrete variable.</p>
CLM Custom	<p>The CLM Custom point has no data other than its CDS blocks, and no function other than its CL programs. Custom points do not have faceplate variables, but do support alarms generated by CL.</p> <p>As an image point, it can be used for CL input, output, and alarm of its CDS variables.</p>
CLM Switch	<p>The CLM Switch point is used for discrete signals and is used to represent digital input, output, and composite type points. Since the CLM does not support a true Digital Composite point, this point uses CL and CDS to emulate the digital composite functionality.</p> <p>As an image point, it can be used for input, output, and alarm of its CDS variables.</p>

Continued on next page

8.1 Standard CLM Point Types and Image Point Selection,

Continued

Point type selection guidelines

Selection of the appropriate CLM data point type to use as an image point depends on an analysis of the type of data or point in the non-Honeywell device.

Table 8-2 Functions to be Considered When Selecting Image Points

Item	Function
1	Non-Honeywell device data/point type.
2	LCN Alarming
3	Display faceplate
4	Input/output control and schedule
5	Data conditioning (CL processing)
6	Other data requirements (support data)

The CLM image point should mirror the function being performed by the non-Honeywell device or controller. In addition, the function should be reflected in a manner familiar to the operator, if possible.

The LCN operator can enter new values for constants in the remote device when appropriate.

Table 8-3 Data Point Type Option Descriptions and Considerations

Point Type Option	Descriptions/Considerations
Input Point Types	These include: <ul style="list-style-type: none">• analog input points,• digital input points,• pulse input points, and• custom or device specific input data points, such as: Smart Transmitters and calculated data points.
Numeric (Analog Input) Point	The CLM numeric point type is used for simple analog inputs. The protocol layer should be configured to store to the PV. There is no Setpoint, mode, output, or CDS.

Continued on next page

8.1 Standard CLM Point Types and Image Point Selection,

Continued

**Non-Honeywell device
data point type,
continued**

Table 8-3 Data Point Type Option Descriptions and Considerations (Continued)

Point Type Option	Description
Analog Input Point	<p>The CLM regulatory point type is used with a PV algorithm of DATAACQ for simple analog inputs.</p> <p>The input connection (PISRC1) parameter P1 should be configured for input from a CDS parameter. The protocol layer should be configured to store to that CDS parameter.</p> <p>SP is an operator-entered value used for deviation alarming.</p> <p>MODE and OP are not used for simple analog inputs. The control algorithm should be null.</p>
Digital Input Point	<p>The CLM flag point type or switch point type can be used for a single digital input.</p> <p>The protocol layer should be configured to store to the PV. Flag points are not scheduled.</p> <p>Storing a value to the PV parameter causes alarm processing.</p>
Output Point Types	<p>The LCN operator interface uses SP, PV, and OP parameters, with the current input state shown in the PV, and the current output in OP.</p> <p>The operator can insert a new output command in the SP parameter. If the mode is MANUAL, the operator can give the OP a new value.</p>
Analog Output Point	<p>Analog output points can be implemented using the CLM regulatory point type, with a null PV algorithm and an auto manual control algorithm.</p> <p>The auto manual algorithm exposes the control output functions, output limits, and so on. In manual mode, the primary task of the auto manual algorithm is to expose CV for user data entry. The final output after output processing is OP.</p> <p>In the cascade mode, CV is computed from X1 plus a bias value.</p>
Digital Output Point	<p>Some digital output points can require a check before output feature. If used, this feature is implemented in the protocol layer which performs the appropriate communication.</p> <p>Digital output points can be included in a digital composite point implemented with the CLM switch algorithm.</p>
Digital Composite Point	<p>Some points may require input monitor and output control. These types may represent pumps on valves and can be implemented using the CLM switch algorithm.</p>
Cascade Control	<p>Note: Cascade control connections to non-Honeywell devices are NOT supported in the CLM interface.</p>

Continued on next page

8.1 Standard CLM Point Types and Image Point Selection,

Continued

LCN alarming

If the input data is to be checked and alarmed on the LCN, then the alarm method can dictate the type of image point to be used.

CLM regulatory points must have a PV algorithm other than NULL to enable PV alarming. The DATAACQ algorithm is normally used for image points. The output of the algorithm is PVCALC. Following PV filtering and range checks, the value is called PVAUTO. PV source selection produces the PV, which is then processed for PV alarming.

PV filtering, source selection and alarm processing can be configured with the DATAACQ algorithm. The control algorithm must be null, auto manual, or CL. General outputs and alarm distribution are available for user configuration.

Display faceplate parameters

The image point data presented to the operator should be in a familiar format, with the display initiated in a familiar way, and with the correct interactions with the operator's keyboard, if possible. Typical faceplate parameters are shown in Table 8-4.

Table 8-4 Typical Faceplate Parameters

Parameter	Description	Keyboard Support
PV	Process Variable	
SP	Setpoint	SP and four triangle keys
OP	Output	OP and four triangle keys
MODE	Mode	MAN, AUTO and NORM keys
	Point status	None

Only the CLM regulatory algorithm faceplate supports all the above parameters. Faceplate parameters for a switchpoint are shown below.

In a regulatory point, the PV receives its value indirectly by an input connection from a CDS parameter.

Setpoint and output are control-mode dependent and are usually handled by CL code.

Setpoint is available for operator entry. Output is determined by CL code.

Although the mode parameter appears on the CDS algorithm, it does not function from the keyboard MODE keys.

Input/output control and schedule

The rate at which the data is input and output, along with the method, is a function of the communication protocol layer.

Digital image points are usually not scheduled by the CLM point processor. Instead, image points are processed when data is received from the non-Honeywell device and stored to the image point. This can be the result of a scheduled PIT/PITG or unsolicited/exception inputs.

Continued on next page

8.1 Standard CLM Point Types and Image Point Selection,

Continued

Display faceplate switchpoint parameters

Table 8-5 Typical Faceplate Parameters for a Switch Point

Parameter	Description
S1REQSTS	SWT 1 Requested State
S1CURSTS	SWT 1 Current State
S2REQSTS	SWT 2 Requested State
S2CURSTS	SWT 2 Current State

Data conditioning (CL processing)

Some data must be preconditioned before it is input to, or output from, the system.

The CLM regulatory, switch, and custom points can be extended by the addition of CDS and CL processing blocks.

Additional parameters unique to a non-Honeywell device are placed in a CDS.

Special point processing, not handled by the protocol layer, can be implemented with an appropriate CL processing block.

This technique is used to implement non-Honeywell points such as:

- Smart transmitter point,
- Tank points,
- Analyzer points,
- Flow calculating, and
- Composite digital points.

Digital composite points are implemented as switch 1 of the CLM switch type.

Alarming implementation requires a CL block at the general insertion point.

Other data requirements (support data)

Some data values can require support status or state information, which may require a CDS block as part of the data interface.

Only the CLM regulatory, switch, and custom points can be extended by the addition of CDS and CL processing blocks. Additional parameters unique to a non-Honeywell device are placed in a CDS.

8.2 CLM Platform Standard Image Point Types

8.2.1 Supported CLM Image Point Types

Supported CLM Platform image point types

Many types of CLM image points are supported as part of the standard CLM Platform software package. The following subsections, listed in Table 8-6, cover these image point types in detail.

Table 8-6 Supported CLM Image Point Types

Subsection	CLM Image Point Type
8.2.2	CLM Digital Image Point Type
8.2.2.3	Digital Composite Point, 3-state Latched
8.2.2.4	Digital Composite Point, 3-state Pulsed
8.2.2.5	Digital Input Point Type, 3-state
8.2.2.6	Digital Output Point, 3-state Latched
8.2.2.7	Digital Output Point, 3-state Pulsed
8.2.2.8	Digital Composite Point, 4-state Latched
8.2.2.9	Digital Composite Point, 4-state Pulsed
8.2.2.10	Digital Input Point, 4-state
8.2.2.11	Digital Output Point, 4-state Latched
8.2.2.12	Digital Output Point, 4-state Pulsed
8.2.2.13	Digital Composite Point, 5-state Latched
8.2.2.14	Digital Composite Point, 5-state Pulsed
8.2.2.15	Digital Output Point, 5-state Latched
8.2.2.16	Digital Output Point, 5-state Pulsed
8.2.3	Flag Image Point Type
8.2.4	Numeric Image Point Type
8.2.5	Pulse Input Image Point Type

8.2.2 CLM Digital Image Point Type

Digital Image Points

The CLM Digital points are based on the CLM Switch type point configured with three, four, or five switch states. These state names are configured by the user to best represent the specific states of the digital device. Custom Data Segment (CDS) packages are then attached to the image point to house data signals and configuration information for the particular Digital point type. Finally, a high-level Control Language (CL) program is linked to the image point to manipulate the digital input and/or output signals, verify proper configuration, and process specific alarm conditions.

Introduction

This section describes in detail the configuration and use of the Digital point type in the CLM. The CLM Digital point is common across all CLM applications that support a Digital point. There are two parts to the Digital point configuration; the image point and the device signal mapping. This manual deals *only* with the configuration and use related to the image point. The input and output signal mapping is specific to the CLM application and is described within that particular manual.

There are eight basic Digital point types supported in the CLM. These types span a varying range of input and output configurations. These eight types are defined by the I/O and the number of states applicable to the particular digital data being represented. The eight Digital point types are:

- ⇒ Dig Input 3-State *or DIN3* (On, Off, Bad)
- ⇒ Dig Input 4-State *or DIN4* (Open, Closed, Moving, Bad)
- ⇒ Dig Composite 3-State *or DIG3* (Start, Stop, Bad)
- ⇒ Dig Composite 4-State *or DIG4* (Open, Close, Moving, Bad)
- ⇒ Dig Composite 5-State *or DIG5* (Open, Close, Stop, Moving, Bad)
- ⇒ Dig Output 3-State *or DOUT3* (Start, Stop, Bad)
- ⇒ Dig Output 4-State *or DOUT4* (Open, Close, Moving, Bad)
- ⇒ Dig Output 5-State *or DOUT5* (Open, Close, Stop, Moving, Bad)

Digital Signals

CLM Digital points merge several digital signals (input and/or output) into a single CLM image point. Digital Input points can monitor one or two input signals to determine a specific state of a device (e.g. On, Off, Open, Closed, etc.). Digital Composite points can drive up to three digital signals to command a device to a specific state (e.g. Start, Stop, Open, Close, etc.) and monitor input and feedback signals to determine that the command was successful.

Alarms are issued by the image point for various process conditions, command failures, and invalid point configurations.

Continued on next page

“BAD” State

The BAD state on the CLM Digital point is used to represent an unknown condition of the digital signals. The BAD state is *always* the last state on the switch point (S1STATES(n) standard parameter) and is treated specially. The CLM Digital point transitions to the BAD state whenever any of the following conditions occur:

- The input signals read from the digital device correspond to the user configured BAD state as determined by the D2D1_{nn} custom parameter on the image point. In this condition, the Switch 1 alarm is On with “BADSTATE” descriptive text.
- The digital device has placed logically Out of Service by the Operator from the CLM Status Display. In this condition, no Switch alarms are issued, but the Switch 2 descriptive text reads “OUT_SERV”.
- The configuration parameters in the CDS of the image point have invalid values. In this condition, Switch 2 alarm is On with “CFGERR_n” descriptive text. *Refer to Table 8-12 for specifics.*
- The input/output component (e.g. board, card, etc.) from which the input or output signals are derived has failed. In this condition, the Switch 2 alarm is On with “PT_FAIL” descriptive text. The PT_FAIL custom parameter is also set to On. *This function is application specific and therefore not used in all applications.*
- For composite points configured with no inputs (e.g. Digital Output points), the output feedback signals do not match the outputs configured for the commanded state (custom parameters ST_n_OP_n). In this condition, Switch 1 alarm is On with “BADSTATE” descriptive text.

Digital Composite I/O

CLM Digital composite points support a configurable number of digital input and output signals. Table 8-7 lists the maximum number of input and output signals that can be configured for each Digital composite point type.

Table 8-7 Max I/O Configuration for Digital Composite

Number of States	Maximum Inputs	Maximum Outputs
3	1	2
4	2	2 or 3 (see table 8-11)
5	2	3

Digital composite points map command states to digital output parameters (OP_n) and input states to digital input parameters (D_n). This configuration is defined through custom parameters on the image point (D2D1_{nn} and D1_n). When the Operator commands a state change from the Digital composite point (S1REQSTS), the CL determines the specific digital output signals required for the commanded state (ST_n_OP_n) and sends them to the Digital device. When input signals arrive from the Digital device, the image point again executes the CL to determine how the input signal(s) map to the input states configured (S1CURSTS).

Continued on next page

8.2.2

CLM Digital Image Point Type, continued

Point type vs. I/O

The CLM Digital point type to be used is determined by the number of input and output signals to be configured. Table 8-8 cross-references input and output signals to a specific Digital point type with the maximum number of commanded states available for that point type.

Table 8-8 Digital Point type vs. I/O

Number of Inputs	Number of Outputs	Digital Point Type	Max Cmd States
0	1	DIG3	2
0	2	DIG3	2
0	2	DIG4	2
0	3	DIG4	2
0	3	DIG5	3
1	0	DIN3	N/A
1	0	DIG3	0
1	1	DIG3	2
1	2	DIG3	2
1	3	DIG4	2
2	0	DIN4	N/A
2	0	DIG4	0
2	1	DIG4	2
2	2	DIG4	2
2	3	DIG4	2
2	3	DIG5	3

Examples

- ⇒ Pump motor control - Requires separate start and stop control signals and single line feedback for running state. Configures as 2 outputs and 1 input. This Digital device would use a DIG3 point type.
- ⇒ Simple Valve control (w/ no STOP) - Requires separate open and close control signals and two feedback signals from the limit switches. Configures as 2 outputs and 2 inputs. This Digital device would use a DIG4 point type.
- ⇒ Advanced Valve control (w/STOP) - Requires separate open, close, and stop control signals and two feedback signals from the limit switches. Configures as 3 outputs and 2 inputs. This Digital device would use a DIG5 point type.
- ⇒ PLC Input Status - Requires a single input signal from the PLC for monitor of the Digital status. Configures as 0 outputs and 1 input. This status would require a DIN3 point type. (A DIG3 with 0 outputs may also be used here.)

Continued on next page

Enumerations on the Digital Points

Valid values for several of the parameters on the CLM Digital point designate the value as \$ord0-\$ord4. These ordinal values correspond to the switch point parameters S1STATES(0)-S1STATES(4). The state parameters (S1REQSTS, S1CURSTS) on the Digital point are parameters of type self-defining enumeration. The name associated with each enumeration value for the state parameters is defined at point build time by the user and assigned to one of the ordinal states using this generic form.

Example:

On a 3-state Digital point,

ON	=	\$ord0	=	S1STATES(0)
OFF	=	\$ord1	=	S1STATES(1)
BAD	=	\$ord2	=	S1STATES(2)

Latched vs. Timed Outputs

The CLM Digital Composite points send outputs to the Digital device and normally expect the output signal to “latch”, or hold at the commanded logical state. As part of the output verification in the attached CL, the feedback signals (FB_n) are read and checked against the outputs sent to the device. In some instances, the digital device pulses the output signal for a specific time duration and then returns to the uncommanded (UNC_OP_n) or “stable” state. In this case, the CLM Digital Composite points need to be configured for timed outputs ($TIMEOP=ON$). Output verification is then checked by comparing the feedback signals (FB_n) against the uncommanded output states (UNC_OP_n). Note that on CLM Digital points configured for pulsed outputs, the physical device (PLC, RTU, etc.) is responsible for resetting the output signals back to their “stable” state.

Important! The readback time field (RBTIME) configured in the associated template mapping file (PITE) must be of a long enough duration to allow for the device output pulse duration to complete. For example, an output pulse duration of 4 seconds should have a readback time greater than 4 seconds.

Continued on next page

8.2.2

CLM Digital Image Point Type, continued

Sending the Outputs

The CLM Digital point is able to send the configured output signals due to what is called the “Store Intercept”. This intercept is a hook into the Application Module data store path which allows the CLM executable tasks to determine that the particular store is for the Digital device. When this store is “intercepted”, the CLM tasks first store to the INITLIZD parameter to set it ON and then complete the store to the SIREQSTS commanded by the Operator. By storing to the SIREQSTS parameter, the AM initiates a PPS of the point which allows the CL attached to the point to execute. The CL then determines the correct output signal configuration to send to the Digital device. The CLM_OUT parameter on the point is set by the CL to initiate the send.

ATTENTION!

If CLM Digital Composite points are to be used in control schemes where CL blocks command the store through the SIREQSTS parameter, the code must first set the INITLIZD parameter to ON. This code would read as follows:

```
--  
-- Start up the pump. Be sure to set the  
-- INITLIZD parameter prior to the store.  
--  
  
SET DIGPOINT.INITLIZD = ON  
SET DIGPOINT.SIREQSTS = START
```

8.2.2.1 CLM Digital Point Alarm Definition

Digital Point Alarming

Alarms on the Digital points are generated in response to abnormal process or configuration conditions. Some of the alarms can be configurable (i.e. the alarm can be inhibited when an abnormal condition occurs). The AM Switch point allows up to three separate alarms to be active simultaneously. Each of these three alarms (SALMFL n) has an eight character descriptor (SALMDSC n) that is generated by the attached CL code to describe the specific alarm condition. Table 8-9 lists the possible alarms that may be generated by a Digital point type.

Table 8-9 Digital Point Alarming Definition

Alarm Text	Switch	Alarm Description																														
OUT_SERV	SALMFL2	<p>ASSOCIATED DEVICE ID OUT OF SERVICE</p> <p>When the communication state of the associated device is either logically Out of Service or in a Hard-Fail condition.</p> <p>NOTE: This condition does not actually generate an alarmable state (i.e. SALMFL2 = OFF; not highlighted).</p>																														
PT_FAIL	SALMFL2	<p>DEVICE POINT OR BOARD FAILURE</p> <p>When the communication to the device is OK, but the device is reporting a problem with the I/O board on which a referenced digital signal exists.</p> <p>NOTE: This alarm is totally device dependent and all applications may not support it.</p>																														
CFGERR_ n	SALMFL2	<p>DIGITAL POINT CONFIGURATION ERROR</p> <p>When the Digital point is set active, the CL will make certain checks for the configuration of the input and output parameters. If it detects a bad configuration, one of these errors will be issued.</p> <table><tr><td>When n =</td><td>0</td><td>NODINPTS value is incorrect</td></tr><tr><td></td><td>1</td><td>NODOPTS value is incorrect</td></tr><tr><td></td><td>2</td><td>ALMOPT value is incorrect</td></tr><tr><td></td><td>3</td><td>NUMCMDST value is incorrect</td></tr><tr><td></td><td>4</td><td>NODOPTS vs. NUMCMDST is incorrect</td></tr><tr><td></td><td>5</td><td>NODINPTS and/or NODOPTS is incorrect for ALMOPT value selected</td></tr><tr><td></td><td>6</td><td>S1NSTATES is incorrect for Digital type</td></tr><tr><td></td><td>7</td><td>NODINPTS vs. NODOPTS is incorrect</td></tr><tr><td></td><td>8</td><td>FBTIME value is incorrect</td></tr><tr><td></td><td>9</td><td>STOP STATE value is incorrect</td></tr></table>	When n =	0	NODINPTS value is incorrect		1	NODOPTS value is incorrect		2	ALMOPT value is incorrect		3	NUMCMDST value is incorrect		4	NODOPTS vs. NUMCMDST is incorrect		5	NODINPTS and/or NODOPTS is incorrect for ALMOPT value selected		6	S1NSTATES is incorrect for Digital type		7	NODINPTS vs. NODOPTS is incorrect		8	FBTIME value is incorrect		9	STOP STATE value is incorrect
When n =	0	NODINPTS value is incorrect																														
	1	NODOPTS value is incorrect																														
	2	ALMOPT value is incorrect																														
	3	NUMCMDST value is incorrect																														
	4	NODOPTS vs. NUMCMDST is incorrect																														
	5	NODINPTS and/or NODOPTS is incorrect for ALMOPT value selected																														
	6	S1NSTATES is incorrect for Digital type																														
	7	NODINPTS vs. NODOPTS is incorrect																														
	8	FBTIME value is incorrect																														
	9	STOP STATE value is incorrect																														

Continued on next page

8.2.2.1 CLM Digital Point Alarm Definition, continued

Table 8-9 Digital Point Alarming Definition, continued

Alarm Text	Switch	Alarm Description
INVCMD	SALMFL3	INVALID COMMAND CHANGE REQUESTED The operator attempted to command an invalid state. Attempting to change to BAD or ILLEGAL, or to an ordinal state that is higher than the number of commanded states allowed (NUMCMDST) are usual causes for this alarm condition.
CMDFAIL	SALMFL1	COMMANDED REQUEST FAILED Either the output signals that correspond to the commanded state were never received by the Digital device, or output signals sent to a pulsed output device never returned to the expected resting state.
CMDDIS	SALMFL1	COMMAND DISAGREE Output feedback signals indicate that the output was received by the Digital device, but input signals received back indicate that the device did not change to the commanded state in the time expected (FBTIME).
UNC_CHG (ALMUNCHG = ON)	SALMFL3	STATE CHANGED WITHOUT OPERATOR COMMAND An uncommanded change of state has been received from the Digital device. NOTE: This alarm is configurable by the user through the ALMUNCHG parameter. When ALMUNCHG is ON, the alarm will be issued; when OFF, the text will appear, but the alarm will not be issued.
OFF_NORM (ALMOPT = 1 or 11)	SALMFL1	STATE DOES NOT MATCH THE NORMAL STATE The current state of the Digital device does not match the configured normal state (PVNORMAL). NOTE: Some states on Digital Input points using two inputs (DIN4) can be configured to only issue Off Normal alarms on switch limit configurations (i.e. D2D1_01 and D2D1_10). See the ALMOPT description.
BAD_PV	SALMFL2	PROCESS VARIABLE IS BAD The feedback from the output values indicate that the outputs are in a BAD state or a configuration error has occurred.

Continued on next page

8.2.2.1 CLM Digital Point Alarm Definition, continued

Table 8-9 Digital Point Alarming Definition, continued

Alarm Text	Switch	Alarm Description
BADSTATE	SALMFL1	DIGITAL POINT IS IN THE BAD STATE The imaged Digital point is in the BAD state. Refer to the description earlier in this section on how the BAD state occurs.
NO_MOVE	SALMFL1	NO MOVEMENT AFTER OPERATOR COMMAND On Digital points using two inputs, feedback signals were never received to indicate the transition state to MOVING or INBETWEEN. NOTE: On Digital Composite points, the timeout checks for this alarm are a configurable portion of the total feedback transition time. Refer to the FBTIME description.
CHGSTATE (ALMOPT = 2 or 12)	SALMFL3	INPUT STATE HAS CHANGED The input signals indicate a change of state in the Digital device. NOTE1: Some states on Digital Input points using two inputs (DIN4) can be configured to only issue Change-of-State alarms on switch limit configurations (i.e. D2D1_01 and D2D1_10). See the ALMOPT description. NOTE2: This alarm issues and then immediately clears, so the Alarm Summary Display will always show this alarm in the Return-to-Normal condition.

8.2.2.2 CLM Digital Point Parameter Definition

Parameter Definition

CLM Digital points use a combination of standard AM Switch parameters and Custom parameters attached to the image point to both configure and process the Digital input and output signals. Table 8-10 below lists these parameters. Many of these parameters are used only by the attached CL block and therefore no entry is required by the user.

Table 8-10 Digital Point Parameter Definitions

Parameter	Valid Values	User Entry	Description
S1NSTATE	3-5	Yes	NUMBER OF SWITCH #1 STATES This parameter defines the number of logical states that will be mapped to the Digital input and output signals.
S1STATE(0)	String	Yes	SWITCH #1 ORDINAL STATE 0 This parameter contains descriptive text to define the ordinal 0 state (or top box on the Detail Display) that maps to the Digital input signals. See the D1_ <i>n</i> and D2D1_ <i>nn</i> parameter descriptions.
S1STATE(1)	String	Yes	SWITCH #1 ORDINAL STATE 1 This parameter contains descriptive text to define the ordinal 1 state (or second box on the Detail Display) that maps to the Digital input signals. See the D1_ <i>n</i> and D2D1_ <i>nn</i> parameter descriptions.
S1STATE(2)	String	Yes	SWITCH #1 ORDINAL STATE 2 This parameter contains descriptive text to define the ordinal 2 state (or third box on the Detail Display) that maps to the Digital input signals. See the D1_ <i>n</i> and D2D1_ <i>nn</i> parameter descriptions.
S1STATE(3)	String	Yes	SWITCH #1 ORDINAL STATE 3 (4-state, 5-state only) This parameter contains descriptive text to define the ordinal 3 state (or fourth box on the Detail Display) that maps to the Digital input signals. See the D1_ <i>n</i> and D2D1_ <i>nn</i> parameter descriptions.
S1STATE(4)	String	Yes	SWITCH #1 ORDINAL STATE 4 (5-state only) This parameter contains descriptive text to define the ordinal 4 state (or fifth box on the Detail Display) that maps to the Digital input signals. See the D1_ <i>n</i> and D2D1_ <i>nn</i> parameter descriptions.

Continued on next page

8.2.2.2 CLM Digital Point Parameter Definition, continued

Table 8-10 Digital Point Parameter Definitions, continued

Parameter	Valid Values	User Entry	Description
S1ACCLVL	Operator Supervisor Engineer Program	Yes	SWITCH #1 COMMAND ACCESS LEVEL This parameter defines the access level required to issue a command change on a Digital point with output signals defined. If using the PROGRAM access level, refer to the earlier discussion on Sending the Outputs.
S1CURSTS	String	Yes	SWITCH #1 CURRENT STATE This standard Switch point parameter defines the current state of the Digital point, as mapped by the input signals. Requires an initial state entry at build time.
S1REQSTS	String	Yes	SWITCH #1 REQUESTED STATE This standard Switch point parameter defines the requested state change of the Digital point, as mapped by the output signals. Requires an initial state entry at build time.
S2NSTATE	2	Yes	NUMBER OF SWITCH #2 STATES This parameter is must be configured as 2. Other values will result in a configuration error alarm.
S2STATE(0)	“TRACK” (DIN3-4) “CLEAR” (DIG3-5)	Yes	SWITCH #2 ORDINAL STATE 0 This string parameter contains descriptive text to define the ordinal 0 state of the second switch.
S2STATE(1)	“NOTRACK” (DIN3-4) “NOACTION” (DIG3-5)	Yes	SWITCH #2 ORDINAL STATE 1 This string parameter contains descriptive text to define the ordinal 1 state of the second switch.
S2ACCLVL	Operator Supervisor Engineer Program	Yes	SWITCH #2 ACCESS LEVEL This parameter defines the access level required to set the TRACK/NOTRACK option on Digital Input points or the CLEAR/NOACTION option on Digital Composite points.
S2CURSTS	String	Yes	SWITCH #2 CURRENT STATE This standard Switch point parameter defines the current state of switch #2 on the Digital point. Requires an initial state entry at build time.
S2REQSTS	String	Yes	SWITCH #2 REQUESTED STATE This standard Switch point parameter defines the requested state change of switch #2 on the Digital point. Requires an initial state entry at build time.

Continued on next page

8.2.2.2 CLM Digital Point Parameter Definition, continued

Table 8-10 Digital Point Parameter Definitions, continued

Parameter	Valid Values	User Entry	Description
D1		No	DIGITAL INPUT SIGNAL - BIT 1 This parameter is used ONLY by the CL.
D2		No	DIGITAL INPUT SIGNAL - BIT 2 This parameter is used ONLY by the CL.
NODINPTS	0-2	Yes	NUMBER OF DIGITAL INPUT SIGNALS This parameter defines the number of Digital device input signals to be processed by the CL for mapping to S1CURSTS.
D1_0	\$ord0-2	Yes	MAPPED STATE WHEN D1=0 This parameter defines the ordinal value for the S1CURSTS when the input signal D1 is OFF. Applies only when there is 1 input signal (NODINPTS=1).
D1_1	\$ord0-2	Yes	MAPPED STATE WHEN D1=1 This parameter defines the ordinal value for the S1CURSTS when the input signal D1 is ON. Applies only to 3-state points when there is 1 input signal (NODINPTS=1).
D2D1_00	\$ord0-4	Yes	MAPPED STATE WHEN D2=0 AND D1=0 This parameter defines the ordinal value for the S1CURSTS when the input signals show D1 and D2 are both OFF. Applies only to 4 and 5-state points when there are 2 input signals (NODINPTS=2).
D2D1_01	\$ord0-4	Yes	MAPPED STATE WHEN D2=0 AND D1=1 This parameter defines the ordinal value for the S1CURSTS when the input signals show D1 as ON and D2 as OFF. Applies only to 4 and 5-state points when there are 2 input signals (NODINPTS=2).
D2D1_10	\$ord0-4	Yes	MAPPED STATE WHEN D2=1 AND D1=0 This parameter defines the ordinal value for the S1CURSTS when the input signals show D1 as OFF and D2 as ON. Applies only to 4 and 5-state points when there are 2 input signals (NODINPTS=2).
D2D1_11	\$ord0-4	Yes	MAPPED STATE WHEN D2=1 AND D1=1 This parameter defines the ordinal value for the S1CURSTS when the input signals show D1 and D2 are both ON. Applies only to 4 and 5-state points when there are 2 input signals (NODINPTS=2).

Continued on next page

8.2.2.2 CLM Digital Point Parameter Definition, continued

Table 8-10 Digital Point Parameter Definitions, continued

Parameter	Valid Values	User Entry	Description
NUMCMDST	1-3	Yes	<p>NUMBER OF COMMANDABLE STATES</p> <p>This parameter defines the number of states that can be changed by the user from S1REQSTS. The value corresponds to the lowest S1STATE(<i>n</i>-1) that can be commanded for output. This means that S1STATE(0) through S1STATE(2) are the only valid output states.</p>
ALMOPT	0-2,11-12	Yes	<p>ALARM OPTIONS:</p> <p>0 - <i>Operational alarms only.</i></p> <p>COMMAND DISAGREE - Alarm when S1CURSTS <> S1REQSTS after FBTIME has elapsed on an operator command change.</p> <p>COMMAND FAILURE - Alarm when the <i>LATCHED</i> feedback values (FB<i>n</i>) from the output do not match the commanded output values (Op<i>n</i>) after FBTIME has elapsed on an operator command change,</p> <p style="text-align: center;"><i>or</i></p> <p>Alarm when the <i>PULSED</i> feedback values (Fb<i>n</i>) do not match the uncommanded output values (UNC_OP<i>n</i>) after FBTIME has elapsed on an operator command change.</p> <p>NO MOVEMENT(4 and 5 state) - Alarm when the S1CURSTS has not changed after the first timeout (fractional part of FBTIME) has elapsed on an operator command change. (See FBTIME definition)</p> <p>1 - <i>Add Off-Normal Alarms.</i></p> <p>Alarm when S1CURSTS changes and S1CURSTS <> PVNORMAL</p> <p>11 - <i>Add Off-Normal Alarms (limits ONLY on 4-5 state)</i></p> <p>Alarm only when the state changes off of normal to another limit (i.e. D2D1_01 or D2D1_10).</p> <p>2 - <i>Change of State Alarms.</i></p> <p>Alarm when S1CURSTS changes (Applies ONLY to Digital Inputs and 3 and 4 state composites configured for NODOPTS = 0)</p> <p>12 - <i>Change of State Alarms (limits ONLY on 4-5 state)</i></p> <p>Alarm only when the state transitions to a limit (i.e. D2D1_01 or D2D1_10).</p>

Continued on next page

8.2.2.2 CLM Digital Point Parameter Definition, continued

Table 8-10 Digital Point Parameter Definitions, continued

Parameter	Valid Values	User Entry	Description
ALMUNCHG	On or Off	Yes	<p>ALARM UNCOMMANDED STATE CHANGES</p> <p>When ON, this parameter causes an alarm when the</p> <p>D1 or D2 inputs change (<i>dig cmp</i>), or</p> <p>the FB1, FB2, FB3 feedback values differ from the output values OP1, OP2, OP3,</p> <p><i>and</i></p> <p>there was no commanded change through the S1REQSTS by the Operator (<i>dig out</i>).</p> <p>When OFF, uncommanded state changes are still mapped to the S1CURSTS, but are not alarmed.</p> <p>See Table 8-12.</p>
PVNORMAL	\$ord0-4	Yes	<p>EXPECTED S1CURSTS NORMAL STATE</p> <p>This parameter is the expected normal state of the point (i.e., the expected value of S1CURSTS): This value is used when the Off Normal alarming option is on (ALMOPT=1).</p>
TIMEOP	On or Off	Yes	<p>TIMED OUTPUTS</p> <p>This parameter indicates whether the output signals at the Digital device are pulsed (ON) or pulsed (OFF). In this configuration, pulsed or timed outputs are defined as the Digital device setting the signals for a specific duration and then returning to a stable or “resting” state. Latched outputs in the Digital device are set and remain at that state until commanded to change.</p> <p>Caution:</p> <p>The readback of the output feedback signals must allow for the pulse duration of the signal.</p>
CMDCHG		No	<p>OPERATOR CHANGE BEING PROCESSED</p> <p>This parameter is used ONLY by the CL.</p>
ALMTYPE1		No	<p>ALARM TYPE CODE FOR SWITCH #1</p> <p>This parameter is used ONLY by the CL.</p>
ALMTYPE2		No	<p>ALARM TYPE CODE FOR SWITCH #2</p> <p>This parameter is used ONLY by the CL.</p>

Continued on next page

8.2.2.2 CLM Digital Point Parameter Definition, continued

Table 8-10 Digital Point Parameter Definitions, continued

Parameter	Valid Values	User Entry	Description
ALMTYPE3		No	ALARM TYPE CODE FOR SWITCH #3 This parameter is used ONLY by the CL.
FBTIME	AAA.BB where: AAA = 0-999, BB = 0-99, B < A	Yes	<p>COMMAND FEEDBACK TIMEOUT (SECONDS)</p> <p>This parameter is the timeout value, in seconds, for a commanded state change request to complete. The feedback timeout value differs between Digital Composite points with 1 input signal and those with 2.</p> <p>For all composite points (3-5 state), The whole number part of this value (AAA) is the total elapsed number of seconds to wait for the input values to indicate that the commanded state has completed (i.e. S1REQSTS=S1CURSTS). Range = 1-999.</p> <p>For composite points with 2 inputs (4-5 state), The fractional part of this value (BB) is multiplied by 100 to obtain the total elapsed number of seconds to wait for the input values to indicate movement (i.e. S1CURSTS equals the ordinal value in D2D1_00 or D2D1_11). The numeric value of the fractional part of the value must be zero for 3-state composite points and must also be less than the whole number value (i.e. AAA > BB). Range = 0-99.</p> <p>At the end of either of these timeouts, or if the parameter has been setup incorrectly, an alarm will be generated.</p> <p>Note: The two timeouts are not cumulative. This means that the total transition time is the whole value (AAA) in seconds. The “No Movement” fractional part (BB) is a subset of the total time.</p> <p>Example: A valve takes 180 seconds to open or close. If we configure for some extra time in the main timeout to be sure the valve command has completed, we might use a value of 200 seconds. Since we do not want the operator to wait that full time before realizing a potential valve problem, and since the valve should begin its movement within 30 seconds, we would configure a FBTIME value of 200.30. If 30 seconds pass and the input signals do not indicate movement, an alarm (NO MOVE) is issued by the point. If the valve <i>does</i> move, but does not complete its transition in 200 seconds, an alarm (CMDDIS) is issued.</p>

Continued on next page

8.2.2.2 CLM Digital Point Parameter Definition, continued

Table 8-10 Digital Point Parameter Definitions, continued

Parameter	Valid Values	User Entry	Description
INITLIZD		No	<p>INITIALIZED - OPERATOR COMMAND ISSUED</p> <p>This parameter is used by both the CL and the CLM executable tasks. When an operator store is performed, the CLM executable tasks set the value to ON before actually allowing the store to the S1REQSTS parameter. Since a store to the S1REQSTS will initiate a PPS of the point and an execution of the CL code, the INITLIZD parameter will be checked by the CL to determine that the store is operator-based and not just a PPS of the point. This allows for repetitive outputs of the same value to the Digital device.</p> <p>Exception: When changing the S1REQSTS from a CL routine, the code MUST set the INITLIZD parameter to ON immediately prior to changing the S1REQSTS state.</p>
TIMER		No	<p>TIMER</p> <p>This value is used ONLY by the CL.</p>
NODOPTS	0-3	Yes	<p>NUMBER OF DIGITAL OUTPUT SIGNALS</p> <p>This parameter is the number of Digital output signals required by the device</p>
ST0_OP1	On or Off	Yes	<p>ORDINAL STATE 0 - OUTPUT BIT 1</p> <p>This parameter defines the first (LSB) bit of the output signals to be mapped to the ordinal state 0. This parameter must be set up when NODOPTS > 0.</p>
ST0_OP2	On or Off	Yes	<p>ORDINAL STATE 0 - OUTPUT BIT 2</p> <p>This parameter defines the second bit of the output signals to be mapped to the ordinal state 0. This parameter must be set up when NODOPTS > 0.</p>
ST0_OP3	On or Off	Yes	<p>ORDINAL STATE 0 - OUTPUT BIT 3</p> <p>This parameter defines the third bit of the output signals to be mapped to the ordinal state 0. This parameter must be set up when NODOPTS > 0.</p>
ST1_OP1	On or Off	Yes	<p>ORDINAL STATE 1 - OUTPUT BIT 1</p> <p>This parameter defines the first (LSB) bit of the output signals to be mapped to the ordinal state 1. This parameter must be set up when NODOPTS > 0.</p>

Continued on next page

8.2.2.2 CLM Digital Point Parameter Definition, continued

Table 8-10 Digital Point Parameter Definitions, continued

Parameter	Valid Values	User Entry	Description
ST1_OP2	On or Off	Yes	ORDINAL STATE 1 - OUTPUT BIT 2 This parameter defines the second bit of the output signals to be mapped to the ordinal state 1. This parameter must be set up when NODOPTS > 0.
ST1_OP3	On or Off	Yes	ORDINAL STATE 1 - OUTPUT BIT 3 This parameter defines the third bit of the output signals to be mapped to the ordinal state 1. This parameter must be set up when NODOPTS > 0.
ST2_OP1	On or Off	Yes	ORDINAL STATE 2 - OUTPUT BIT 1 This parameter defines the first (LSB) bit of the output signals to be mapped to the ordinal state 2. This parameter must be set up when NODOPTS > 0.
ST2_OP2	On or Off	Yes	ORDINAL STATE 2 - OUTPUT BIT 2 This parameter defines the second bit of the output signals to be mapped to the ordinal state 2. This parameter must be set up when NODOPTS > 0.
ST2_OP3	On or Off	Yes	ORDINAL STATE 2 - OUTPUT BIT 3 This parameter defines the third bit of the output signals to be mapped to the ordinal state 2. This parameter must be set up when NODOPTS > 0.
UNC_OP1	On or Off	Yes	UNCOMMANDED OUTPUT STABLE STATE - BIT 1 This parameter applies ONLY to timed outputs (TIMEOP = ON). This is the stable or "resting" state to which the output signal is expected to return after the pulse duration. This value is compared with the feedback parameter FB1 to determine if the output command was successful.
UNC_OP2	On or Off	Yes	UNCOMMANDED OUTPUT STABLE STATE - BIT 2 This parameter applies ONLY to timed outputs (TIMEOP = ON). This is the stable or "resting" state to which the output signal is expected to return after the pulse duration. This value is compared with the feedback parameter FB2 to determine if the output command was successful.
UNC_OP3	On or Off	Yes	UNCOMMANDED OUTPUT STABLE STATE - BIT 3 This parameter applies ONLY to timed outputs (TIMEOP = ON). This is the stable or "resting" state to which the output signal is expected to return after the pulse duration. This value is compared with the feedback parameter FB3 to determine if the output command was successful.

Continued on next page

8.2.2.2 CLM Digital Point Parameter Definition, continued

Table 8-10 Digital Point Parameter Definitions, continued

Parameter	Valid Values	User Entry	Description
OP1		No	DIGITAL OUTPUT SIGNAL - BIT 1 This parameter is used ONLY by the CL.
OP2		No	DIGITAL OUTPUT SIGNAL - BIT 2 This parameter is used ONLY by the CL.
OP3		No	DIGITAL OUTPUT SIGNAL - BIT 3 This parameter is used ONLY by the CL.
FB1		No	DIGITAL OUTPUT FEEDBACK SIGNAL - BIT 1 This parameter is used ONLY by the CL.
FB2		No	DIGITAL OUTPUT FEEDBACK SIGNAL - BIT 2 This parameter is used ONLY by the CL.
FB3		No	DIGITAL OUTPUT FEEDBACK SIGNAL - BIT 3 This parameter is used ONLY by the CL.
S1REQOLD		No	PREVIOUS REQUESTED OUTPUT STATE This parameter is used ONLY by the CL.
S1REQLST		No	LAST OPERATOR COMMAND CHANGE REQUEST This parameter is used ONLY by the CL.
S1CUROUT		No	CURRENT CALCULATED OUTPUT STATE This parameter is used ONLY by the CL.
UNCLRTIM	-1, 0, 1- <i>n</i>	Yes	CONFIGURE UNCOMMANDED CHANGE ALARM CLEAR This parameter is the configuration field for clearing uncommanded change alarms, if ALMUNCHG=ON. -1 = Do not automatically clear the alarm. 0 = Clear the alarm the next time the point executes (by PPS or PIT update). 1- <i>n</i> = Wait <i>n</i> seconds before clearing the alarm.
UNCCLEAR		No	CLEAR THE UNCOMMANDED CHANGE ALARM This parameter is used ONLY by the CL.
STOP_ST	0-2	Yes	ORDINAL VALUE FOR THE STOP STATE This parameter is the ordinal position for the STOP state. This applies ONLY to the 5-state Digital point.
MOVE_ST		No	ORDINAL VALUE FOR THE MOVE STATE This parameter is fixed at 3 and used ONLY by the CL in the 5-state Digital point.
DATA		No	FUTURE

8.2.2.3 Digital Composite Point, 3-State Latched

Operation

The 3 State Digital Composite point emulates digital devices with one input and either one or two outputs (e.g. pumps, motors). The input states (S1CURSTS) reflect an ON, OFF (\$ord0-\$ord1), and BAD or COMM_ERR (\$ord2) condition. The D1 parameter reflects the current logical state of the Digital device, while the OP1 and OP2 parameters correspond to the latched output signals required to drive the Digital device.

A CL block is attached to this image point that:

- ⇒ Maps the D1 input signal to one of the ordinal input states (S1CURSTS).
- ⇒ When changed by the operator, maps the ordinal output state (S1REQSTS) to the OP1 and OP2 output signals, and executes the send to the Digital device.
- ⇒ When an operator change is in progress, monitors the input signal (D1) for the requested state change against the configured time allowed for the transition (FBTIME), and either alarms or sets the ordinal state (S1CURSTS) to reflect the change.
- ⇒ If configured in ALMUNCHG, alarms when an input state change (D1) occurs that was not commanded by the operator.
- ⇒ If configured in the alarm options (ALMOPT), issues Off Normal alarms when the input state (D1) changes off of the normal state (PVNORMAL).

When an operator commands a change (S1REQSTS), the configured output combination is sent to the Digital device. The output feedback signals (FB1, FB2) are checked to verify that the output signals (OP1, OP2) were properly received and processed by the Digital device. The input signal (D1) is then monitored for the commanded state change. If this state change is not detected before the feedback timer (FBTIME) expires, a CMD_FAIL or CMDDIS alarm is issued.

Alarms may be cleared or the states aligned by selecting the secondary switch (S2REQSTS) option for CLEAR. The CL will set the state back to NOACTION when complete.

Application

Digital device type examples:

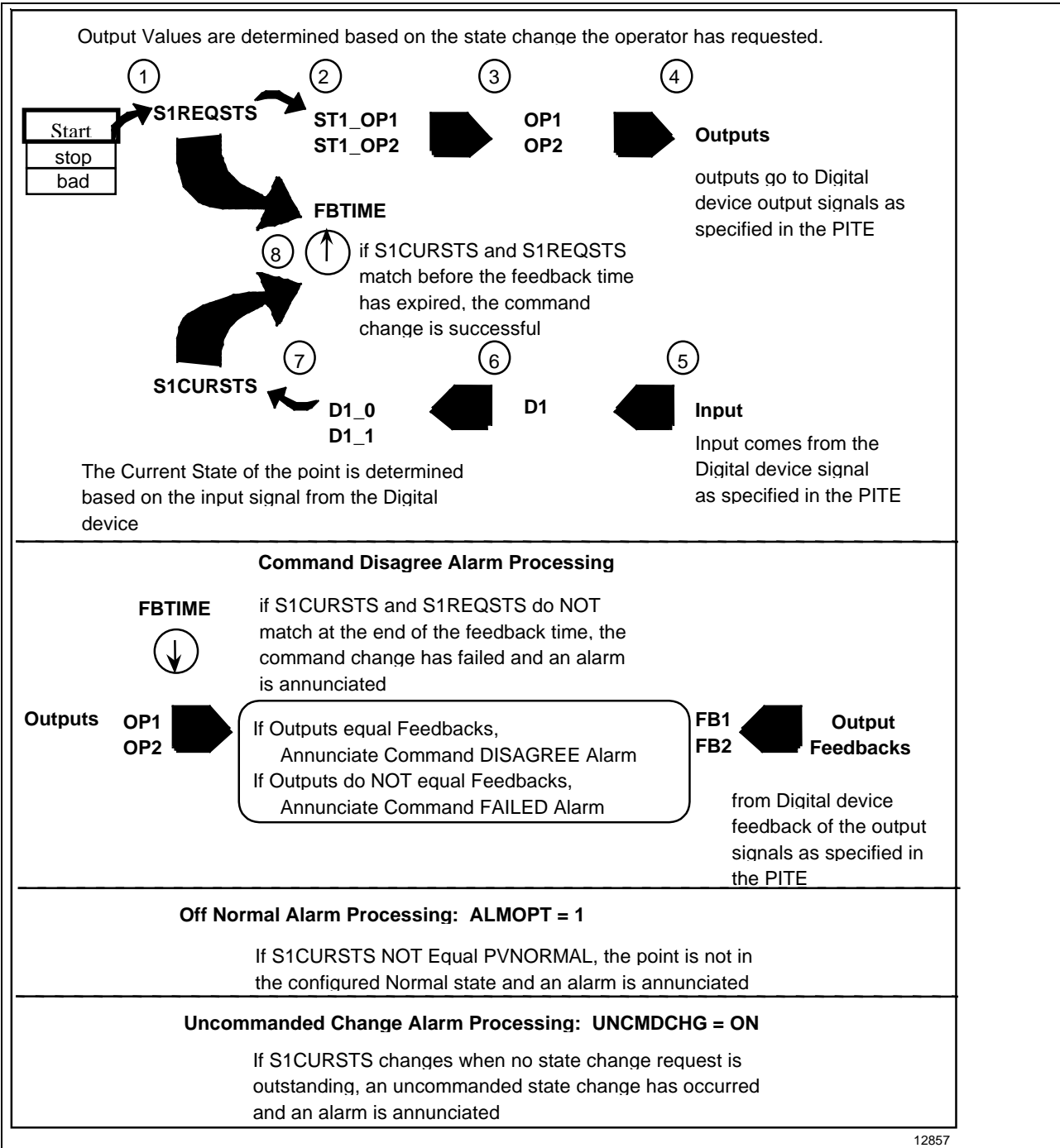
- Pump motor monitor and control
- Fan motor monitor and control
- Devices requiring 1 input and 1 or 2 latched output signals

8.2.2.3 Digital Composite Point - 3-State Latched, continued

Operational Flow

Following is a depiction of how a 3-State digital composite operates. It shows a Digital Composite with two output signals that are **Latched** Outputs with one input signal.

Figure 8-1 Operational Flow for 3-State Latched Digital Composite



8.2.2.3 Digital Composite Point - 3-State Latched, continued

I/O Configuration Options

Number of Inputs	Number of Outputs	Max Cmd States
1	1	2
1	2	2

Alarming

The following is a list of alarms that may be issued by this Digital point type. Refer to Table 8-9 for specific alarm definition.

- OUT_SERV
- PT_FAIL
- CFG_ERR_*n*
- INVCMD
- CMDFAIL
- CMDDIS
- UNC_CHG (If configured through ALMUNCHG)
- OFF_NORM (If ALMOPT = 1)
- BAD_PV
- BADSTATE

Build the Point

The 3-State Digital point is based on the AM Switch point. There are two methods of building this point type. Both of these methods use the Data Entity Builder (DEB) and the data entry requirements listed in Table 8-11 below.

- 1) Select the AM SWITCH point type and enter data as listed below.
or
- 2) Copy the DIG3L.EB Exception Build file (included and installed with the Platform release media) to another file for editing.

Edit the new Exception Build file using the data listed below as input for each of the points to be included within the file. Cut and Paste as required.

Perform an Exception Build process on the new file.

Refer to Table 8-10 for complete descriptions of all Digital point parameters.

See Figure 8-2 for an Exception Build file example.

8.2.2.3 Digital Composite Point - 3-State Latched, continued

Table 8-11 3-State Latched Digital Composite Parameter Entries

Parameter	Entry	Required?	Notes
NAME		Yes	See PKGNAME(4) notes below
UNIT		Yes	
CLSLOTS	1	Yes	
NOPKG	3 or 4	Yes	
NUMSWTCH	2	Yes	
PKGNAME(1)	CLM_DMD3	Yes	
PKGNAME(2)	DIG3CDS	Yes	
PKGNAME(3)	DIG3CDS1	Yes	
PKGNAME(4)	See Notes	-	
S1NSTATE	3	Yes	Enter text for the ordinal 0 (or top) state Enter text for the ordinal 1 (or middle) state Ordinal 2 (or bottom) state text. This state must be the BAD state. Key Access Level for command changes. Default is ENGINEER.
S1STATE(0)	See Notes	Yes	
S1STATE(1)	See Notes	Yes	
S1STATE(2)	BAD	Yes	
S1ACCLVL	Operator Supervisor Engineer Program	No	
S2NSTATE	2	Yes	Selected state for "resetting" the point. Default state. Selecting has no impact. Key Access Level for clearing switch alarms and setting commanded state = current state. This setting should match the S1ACCLVL. Default is ENGINEER.
S2STATE(0)	CLEAR	Yes	
S2STATE(1)	NOACTION	Yes	
S2ACCLVL	Operator Supervisor Engineer Program	No	
PERIOD	NOPERIOD	Yes	CLM Digital points are PPS'd for execution by data received by exception input or the CLM Point Interface Table (PIT) scheduler.

Continued on next page

8.2.2.3 Digital Composite Point - 3-State Latched, continued

Table 8-11 3-State Latched Digital Composite Parameter Entries, continued

Parameter	Entry	Required?	Notes
D1_0	\$ord0-\$ord1	Yes	S1CURSTS becomes the ordinal state identified when D1 is OFF.
D1_1	\$ord0-\$ord1	Yes	S1CURSTS becomes the ordinal state identified when D1 is ON.
NUMCMDST	1-2	Yes	Set to 1 if only ordinal state 0 can be commanded. Set to 2 if both ordinal 0 and 1 can be commanded.
ALMOPT	0-1	No	Default = 0 for operational alarms only. Set to 1 to add Off Normal alarming.
ALMUNCHG	ON or OFF	No	Default = OFF. Set to ON to alarm uncommanded state changes.
PVNORMAL	\$ord0-\$ord1	Yes IF ALMOPT=1	Not required when ALMOPT = 0. Required when ALMOPT = 1.
NODINPTS	1	Yes	Number of Digital Input signals is 1.
FBTIME	See Notes	Yes	Feedback Timeout. Maximum time (in secs) allowed for the commanded state change. Refer to Table 8-10.
TIMEOP	OFF	No	Default = OFF for "Latched" outputs.
NODOPTS	1-2	Yes	Number of Digital output signals.
ST0_OP1	ON or OFF	Yes	For ordinal state 0, bit 1. Set to ON for a logical "1", OFF for a logical "0".
ST0_OP2	ON or OFF	Yes IF NODOPTS=2	For ordinal state 0, bit 2. IF NODOPTS = 2, set to ON for a logical "1", OFF for a logical "0".
ST1_OP1	ON or OFF	Yes	For ordinal state 1, bit 1. Set to ON for a logical "1", OFF for a logical "0".
ST1_OP2	ON or OFF	Yes IF NODOPTS=2	For ordinal state 1, bit 2. IF NODOPTS = 2, set to ON for a logical "1", OFF for a logical "0".
UNCLRTIM	-1, 0, or 1- <i>n</i>	Yes IF ALMUNCHG= ON	Configurable Clear of the Uncommanded Change alarm. IF ALMUNCHG = ON, set to: -1 for Operator clear only, 0 for automatic clear on next point execution 1- <i>n</i> for automatic clear after <i>n</i> seconds.

8.2.2.3 Digital Composite Point - 3-State Latched, continued

Build Example

A Digital point needs to be built to emulate pump #1. The pump requires output signal #1 = ON for RUN and output signal #2 = ON for OFF. The state of the pump is monitored through 1 input signal (ON = RUN). It takes approximately 15 seconds for the pump to start and transition to the RUN state. The operator needs to be able to command the pump and wants to know if the pump is started or stopped at the pump (uncommanded change). The interface is Modbus and therefore requires no additional CDS packages. Figure 8-2 below shows an example of an Exception Build file that could be used to create this image point.

Figure 8-2 Build Example for 3-State Latched Digital Composite

```
{SCREEN ENTITY PUMP_1() }
&T SWITCHAM
&N PUMP_1
UNIT = C1
PTDESC ="PUMP 1 CONTROL  "
KEYWORD ="  "
CLSLOTS = 1
NOPKG = 3
NUMSWTCH = 2          {**NOTE: Will cause CL LINK ERROR if not set to 2**}
PRIMMOD = _____
SALMDSC1 ="  "
SALMDSC2 ="  "
SALMDSC3 ="  "
BCLEALPR = LOW        {**NOTE: R500 And Later**}
BCLFALPR = LOW        {**NOTE: R500 And Later**}
CLEALMPR = LOW        {**NOTE: R500 And Later**}
CLFALMPR = LOW        {**NOTE: R500 And Later**}
CNFERRPR = LOW        {**NOTE: R500 And Later**}
SWALM1PR = LOW        {**NOTE: R500 And Later**}
SWALM2PR = LOW        {**NOTE: R500 And Later**}
SWALM3PR = LOW        {**NOTE: R500 And Later**}
ALENBST = ENABLE
PKGNAME(1) ="CLM_DMD3"
PKGNAME(2) ="DIG3CDS "
PKGNAME(3) ="DIG3CDS1"
S1NSTATE = 3
S1STATES(0) ="RUN  "   {**NOTE: User-entered text**}
S1STATES(1) ="OFF  "   {**NOTE: User-entered text**}
S1STATES(2) ="BAD  "
S1ACCLVL = OPERATOR
S1CURSTS = OFF
S1REQSTS = OFF
S2NSTATE = 2
```

Continued on next page

8.2.2.3 Digital Composite Point - 3-State Latched, continued

Figure 8-2 Build Example for 3-State Latched Digital Composite , continued

```
S2STATES(0) ="CLEAR "
S2STATES(1) ="NOACTION"
S2ACCLVL = OPERATOR
S2CURSTS = NOACTION
S2REQSTS = NOACTION
$IPPASN = OFF
PERIOD = NOPERIOD          {**NOTE: Scheduled and driven by associated PIT**}
CLM_PORT =" "
CLM_DEV =" "
CLM_PIT =" "
NODINPTS = 1
D1_0 = $ORD1               {**NOTE: Pump is OFF when state is logical zero**}
D1_1 = $ORD0
NUMCMDST = 2
ALMOPT = 0
ALMUNCHG = ON
PVNORMAL = $ORD1
FBTIME = 20.0
TIMEOP = OFF
NODOPTS = 2
ST0_OP1 = OFF              {**NOTE: RUN state sets output signal #2 to ON**}
ST0_OP2 = ON
ST1_OP1 = ON               {**NOTE: OFF state sets output signal #1 to ON**}
ST1_OP2 = OFF
UNC_OP1 = OFF
UNC_OP2 = OFF
UNCLRTIM = -1
UNCCLEAR = OFF
DEV_STA = OFF
PT_FAIL = OFF
CMDCHG = OFF
ALMTYPE1 = A_NONE
ALMTYPE2 = A_NONE
ALMTYPE3 = A_NONE
INITLIZD = OFF
TIMER = 0
DATA(1) = 0
DATA(2) = 0
DATA(3) = 0
DATA(4) = 0
DATA(5) = 0
DATA(6) = 0
DATA(7) = 0
DATA(8) = 0
DATA(9) = 0
DATA(10) = 0
```

Continued on next page

8.2.2.3 Digital Composite Point - 3-State Latched, continued

Attach the CL Block

All CLM Digital points require a CL Block. The 3-State Digital Composite uses the DIG3CL.AO block included with the CLM Platform release media. Use the EC file included to link the block to the 3-State point.

From the COMMAND PROCESSOR window of the Engineering Personality:

```
EC DIG3.EC pppppppp
      where
      pppppppp is the 3-state point name
```

NOTE: 1) Be sure that your pathnames are set for the CL SOURCE/OBJ to point to the proper directory.
2) When executing this EC file for the first time, it is normal to get an error on the UNLINK since the block has not yet been attached.

Link Caution!

If the number of switches (NUMSWTCH) on the image point is not set to 2, the following error will be issued by the Linker.

Error during attempt to get parameter information; PA Status: END OF USER SPEC

Correct and rebuild the image point before attempting to re-link.

Activate the Point

At the Detail Display of the Digital point, set the PTEXECST = ACTIVE. You may need to PROCESS the point again to clear the RESTART state to NONE. The S1REQSTS and S1CURSTS should agree at this point, but if not, select the S2REQSTS, select the CLEAR option and press ENTER.

Map to the Device (See Application User Manual)

The build of the image is complete from the AM side. The next step is to assign the input and/or output signals to the required parameters. This is done through the CLM Configuration Builder and is application-specific. This means that each application will map to the same image point parameters, but the format and conversion process for each specific application will probably vary. Refer to the earlier sections on the CLM Configuration Builder in this manual and the specific application user manual for instructions on mapping your particular device.

Continued on next page

8.2.2.3 Digital Composite Point - 3-State Latched, continued

Command the Point

The Digital point may command an output through the S1REQSTS parameter of the image point. *THE DIGITAL POINT MUST HAVE BEEN MAPPED ALREADY THROUGH THE CLM CONFIGURATION BUILDER.*

The steps to command an output are:

- Select the S1REQSTS parameter

- Select the desired state

- Press ENTER

This will PPS the point and cause the CL to execute, sending the appropriate OPn parameter values to the device.

<p>IMPORTANT! A mapped Digital point will not complete the S1REQSTS store if the associated device (NDT) is Out of Service or in the HARD-FAIL condition. This may be checked at the point by examining the DEV_STA parameter. If the value is OFF, then S1REQSTS stores will not complete and the operator will not see the requested change at the Detail Display.</p>

Clear the Point

After certain alarms or other conditions, it may become necessary to clear an existing alarm(s) and/or line up the S1REQSTS and S1CURSTS parameters. This can be done through the second switch on the Digital point. The steps to perform this action are:

- Select the S2REQSTS parameter

- Select the CLEAR state

- Press ENTER

This action will have the impact of setting the S1REQSTS = S1CURSTS, clearing all existing switch alarms, and re-issuing new alarms based on current conditions.

8.2.2.4 Digital Composite Point, 3-State Pulsed

Operation

The 3 State Digital Composite point emulates digital devices with one input and either one or two outputs (e.g. pumps, motors). The input states (S1CURSTS) reflect an ON, OFF (\$ord0-\$ord1), and BAD or COMM_ERR (\$ord2) condition. The D1 parameter reflects the current logical state of the Digital device, while the OP1 and OP2 parameters correspond to the pulsed output signals required to drive the Digital device.

A CL block is attached to this image point that:

- ⇒ Maps the D1 input signal to one of the ordinal input states (S1CURSTS).
- ⇒ When changed by the operator, maps the ordinal output state (S1REQSTS) to the OP1 and OP2 output signals, and executes the send to the Digital device.
- ⇒ When an operator change is in progress, monitors the input signal (D1) for the requested state change against the configured time allowed for the transition (FBTIME), and either alarms or sets the ordinal state (S1CURSTS) to reflect the change.
- ⇒ If configured in ALMUNCHG, alarms when an input state change (D1) occurs that was not commanded by the operator.
- ⇒ If configured in the alarm options (ALMOPT), issues Off Normal alarms when the input state (D1) changes off of the normal state (PVNORMAL).

When an operator commands a change (S1REQSTS), the configured output combination is sent to the Digital device. The uncommanded output state signals (UNC_OP1, UNC_OP2) are checked to verify that the output signals (OP1, OP2) were properly received and processed by the Digital device. The input signal (D1) is then monitored for the commanded state change. If this state change is not detected before the feedback timer (FBTIME) expires, a CMD_FAIL or CMDDIS alarm is issued.

Alarms may be cleared or the states aligned by selecting the secondary switch (S2REQSTS) option for CLEAR. The CL will set the state back to NOACTION when complete.

Application

Digital device type examples:

- Pump motor monitor and control
- Fan motor monitor and control
- Devices requiring 1 input and 1 or 2 pulsed output signals

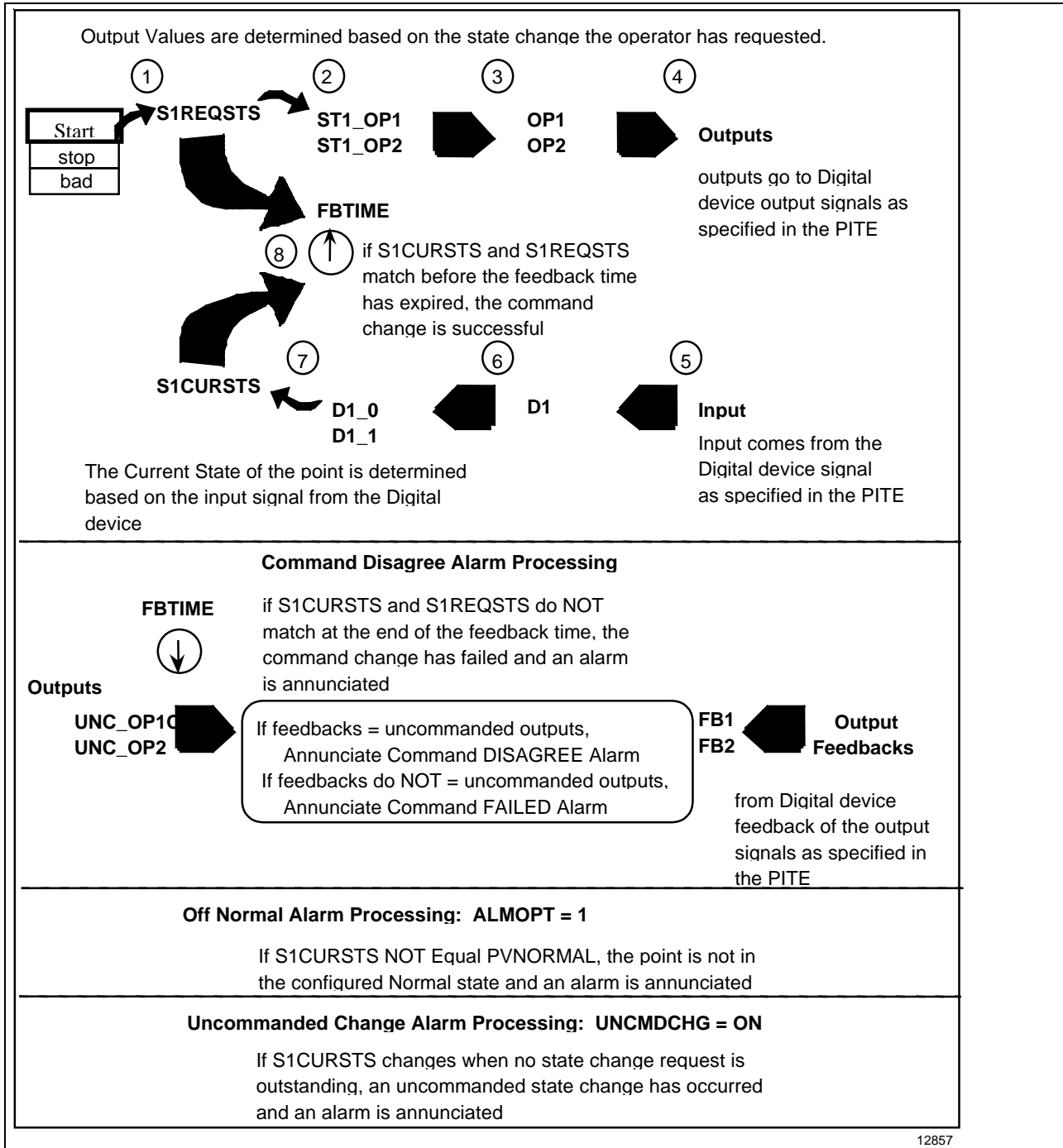
Continued on next page

8.2.2.4 Digital Composite Point - 3-State Pulsed, continued

Operational Flow

Following is a depiction of how a 3-State digital composite operates. It shows a Digital Composite with two output signals that are **Pulsed** Outputs with one input signal.

Figure 8-3 Operational Flow for 3-State Pulsed Digital Composite



Continued on next page

8.2.2.4 Digital Composite Point - 3-State Pulsed, continued

I/O Configuration Options

Number of Inputs	Number of Outputs	Max Cmd States
1	1	2
1	2	2

Alarming

The following is a list of alarms that may be issued by this Digital point type. Refer to Table 8-9 for specific alarm definition.

- OUT_SERV
- PT_FAIL
- CFG_ERR_*n*
- INVCMD
- CMDFAIL
- CMDDIS
- UNC_CHG (If configured through ALMUNCHG)
- OFF_NORM (If ALMOPT = 1)
- BAD_PV
- BADSTATE

Build the Point

The 3-State Digital point is based on the AM Switch point. There are two methods of building this point type. Both of these methods use the Data Entity Builder (DEB) and the data entry requirements listed in Table 8-12 below.

- 1) Select the AM SWITCH point type and enter data as listed below.
or
- 2) Copy the DIG3P.EB Exception Build file (included and installed with the Platform release media) to another file for editing.

Edit the new Exception Build file using the data listed below as input for each of the points to be included within the file. Cut and Paste as required.

Perform an Exception Build process on the new file.

Refer to Table 8-10 for complete descriptions of all Digital point parameters.

See Figure 8-4 for an Exception Build file example.

Continued on next page

8.2.2.4 Digital Composite Point - 3-State Pulsed, continued

Table 8-12 3-State Pulsed Digital Composite Parameter Entries

Parameter	Entry	Required?	Notes
NAME		Yes	See PKGNAME(4) notes below
UNIT		Yes	
CLSLOTS	1	Yes	
NOPKG	3 or 4	Yes	
NUMSWTCH	2	Yes	
PKGNAME(1)	CLM_DMD3	Yes	
PKGNAME(2)	DIG3CDS	Yes	
PKGNAME(3)	DIG3CDS1	Yes	
PKGNAME(4)	See Notes	-	Application Specific - Refer to the User Manual section for any required packages
S1NSTATE	3	Yes	Ordinal 2 (or bottom) state text. This state must be the BAD state.
S1STATE(0)	See Notes	Yes	
S1STATE(1)	See Notes	Yes	
S1STATE(2)	BAD	Yes	
S1ACCLVL	Operator Supervisor Engineer Program	No	
S2NSTATE	2	Yes	Selected state for “resetting” the point.
S2STATE(0)	CLEAR	Yes	
S2STATE(1)	NOACTION	Yes	
S2ACCLVL	Operator Supervisor Engineer Program	No	
PERIOD	NOPERIOD	Yes	CLM Digital points are PPS'd for execution by data received by exception input or the CLM Point Interface Table (PIT) scheduler.

Continued on next page

8.2.2.4 Digital Composite Point - 3-State Pulsed, continued

Table 8-12 3-State Pulsed Digital Composite Parameter Entries, continued

Parameter	Entry	Required?	Notes
D1_0	\$ord0-\$ord1	Yes	S1CURSTS becomes the ordinal state identified when D1 is OFF.
D1_1	\$ord0-\$ord1	Yes	S1CURSTS becomes the ordinal state identified when D1 is ON.
NUMCMDST	1-2	Yes	Set to 1 if only ordinal state 0 can be commanded. Set to 2 if both ordinal 0 and 1 can be commanded.
ALMOPT	0-1	No	Default = 0 for operational alarms only. Set to 1 to add Off Normal alarming.
ALMUNCHG	ON or OFF	No	Default = OFF. Set to ON to alarm uncommanded state changes.
PVNORMAL	\$ord0-\$ord1	Yes IF ALMOPT=1	Not required when ALMOPT = 0. Required when ALMOPT = 1.
NODINPTS	1	Yes	Number of Digital Input signals is 1.
FBTIME	See Notes	Yes	Feedback Timeout. Maximum time (in secs) allowed for the commanded state change. Refer to Table 8-13.
TIMEOP	ON	Yes	Set to ON for "Pulsed" outputs.
NODOPTS	1-2	Yes	Number of Digital output signals.
ST0_OP1	ON or OFF	Yes	For ordinal state 0, bit 1. Set to ON for a logical "1", OFF for a logical "0".
ST0_OP2	ON or OFF	Yes IF NODOPTS=2	For ordinal state 0, bit 2. IF NODOPTS = 2, set to ON for a logical "1", OFF for a logical "0".
ST1_OP1	ON or OFF	Yes	For ordinal state 1, bit 1. Set to ON for a logical "1", OFF for a logical "0".
ST1_OP2	ON or OFF	Yes IF NODOPTS=2	For ordinal state 1, bit 2. IF NODOPTS = 2, set to ON for a logical "1", OFF for a logical "0".

Continued on next page

8.2.2.4 Digital Composite Point - 3-State Pulsed, continued

Table 8-12 3-State Pulsed Digital Composite Parameter Entries, continued

Parameter	Entry	Required?	Notes
UNC_OP1	ON or OFF	Yes	Uncommanded “stable” state for output bit 1. Set to ON for a logical “1”, OFF for a logical “0”.
UNC_OP2	ON or OFF	Yes IF NODOPTS=2	Uncommanded “stable” state for output bit 2. Set to ON for a logical “1”, OFF for a logical “0”.
UNCLRTIM	-1, 0, or 1- <i>n</i>	Yes IF ALMUNCHG= ON	Configurable Clear of the Uncommanded Change alarm. IF ALMUNCHG = ON, set to: -1 for Operator clear only, 0 for automatic clear on next point execution 1- <i>n</i> for automatic clear after <i>n</i> seconds.

Continued on next page

8.2.2.4 Digital Composite Point - 3-State Pulsed, continued

Build Example

The user wants to build a Digital point to emulate pump 1. The pump requires output signal #1 = ON for RUN and output signal #2 = ON for OFF. The device pulses all ON, or logical "1" signals for 4 seconds and then returns back to the OFF, or logical "0" state. The running state of the pump is monitored through 1 input signal (OFF=RUN). It takes approximately 10 seconds for the pump to start and transition to the RUN state. The operator needs to be able to command the pump. The pump cannot be started or stopped at the pump (no uncommanded change). The interface is Bristol RTU and therefore requires an additional CDS package (refer to the Bristol-Babcock RTU Application User Manual for CLM). Figure 8-5 below shows an example of an Exception Build file (in LCN R500 format) that could be used to create this image point.

Figure 8-4 Build Example for 3-State Pulsed Digital Composite {SCREEN ENTITY PUMP_1() }

```
&T SWITCHAM
&N PUMP_1
UNIT = C1
PTDESC ="PUMP 1 CONTROL"
KEYWORD =" "
CLSLOTS = 1
NOPKG = 4
NUMSWTCH = 2 {**NOTE: Will cause CL LINK ERROR if not set to 2**}
PRIMMOD = _____
SALMDSC1 =" "
SALMDSC2 =" "
SALMDSC3 =" "
BCLEALPR = LOW {**NOTE: R500 And Later**}
BCLFALPR = LOW {**NOTE: R500 And Later**}
CLEALMPR = LOW {**NOTE: R500 And Later**}
CLFALMPR = LOW {**NOTE: R500 And Later**}
CNFERRPR = LOW {**NOTE: R500 And Later**}
SWALM1PR = LOW {**NOTE: R500 And Later**}
SWALM2PR = LOW {**NOTE: R500 And Later**}
SWALM3PR = LOW {**NOTE: R500 And Later**}
ALENBST = ENABLE
PKGNAME(1) ="CLM_DMD3"
PKGNAME(2) ="DIG3CDS "
PKGNAME(3) ="DIG3CDS1"
PKGNAME(4) ="B_DG3CDS" {**NOTE: Provided and required by the Interface**}
S1NSTATE = 3
S1STATES(0) ="RUN " {**NOTE: User-entered text**}
S1STATES(1) ="OFF " {**NOTE: User-entered text**}
S1STATES(2) ="BAD "
S1ACCLVL = SUPERVIS
S1CURSTS = OFF
S1REQSTS = OFF
```

Continued on next page

8.2.2.4 Digital Composite Point - 3-State Pulsed, continued

Figure 8-4 Build Example for 3-State Pulsed Digital Composite , continued

```
S2NSTATE = 2
S2STATES(0) ="CLEAR  "
S2STATES(1) ="NOACTION"
S2ACCLVL = OPERATOR
S2CURSTS = NOACTION
S2REQSTS = NOACTION
SIPPASN = OFF
PERIOD = NOPERIOD          {**NOTE: Scheduled and driven by associated PIT**}
CLM_PORT ="      "
CLM_DEV ="      "
CLM_PIT ="      "
NODINPTS = 1
D1_0 = $ORD0               {**NOTE: Pump is ON when state is logical zero**}
D1_1 = $ORD1
NUMCMDST = 2
ALMOPT = 0
ALMUNCHG = OFF
PVNORMAL = $ORD1
FBTIME = 15.0
TIMEOP = ON                {**NOTE: MUST be ON for Pulsed output signals**}
NODOPTS = 2
ST0_OP1 = OFF              {**NOTE: RUN state sets output signal #2 to ON**}
ST0_OP2 = ON
ST1_OP1 = ON               {**NOTE: OFF state sets output signal #1 to ON**}
ST1_OP2 = OFF
UNC_OP1 = OFF
UNC_OP2 = OFF
UNCLRTIM = -1
UNCCLEAR = OFF
DEV_STA = OFF
PT_FAIL = OFF
CMDCHG = OFF
ALMTYPE1 = A_NONE
ALMTYPE2 = A_NONE
ALMTYPE3 = A_NONE
INITLIZD = OFF
TIMER = 0
DATA(1) = 0
DATA(2) = 0
DATA(3) = 0
DATA(4) = 0
DATA(5) = 0
DATA(6) = 0
DATA(7) = 0
DATA(8) = 0
DATA(9) = 0
DATA(10) = 0
```

8.2.2.4 Digital Composite Point - 3-State Pulsed, continued

Attach the CL Block

All CLM Digital points require a CL Block. The 3-State Digital Composite uses the DIG3CL.AO block included with the CLM Platform release media. Use the EC file included to link the block to the 3-State point.

From the COMMAND PROCESSOR window of the Engineering Personality:

```
EC DIG3.EC pppppppp
where
pppppppp is the 3-State point name
```

NOTE: 1) Be sure that your pathnames are set for the CL SOURCE/OBJ to point to the proper directory.
2) When executing this EC file for the first time, it is normal to get an error on the UNLINK since the block has not yet been attached.

Link Caution!

If the number of switches (NUMSWTCH) on the image point is not set to 2, the following error will be issued by the Linker.

Error during attempt to get parameter information; PA Status: END OF USER SPEC

Correct and rebuild the image point before attempting to re-link.

Activate the Point

At the Detail Display of the Digital point, set the PTEXECST = ACTIVE. You may need to PROCESS the point again to clear the RESTART state to None. The S1REQSTS and S1CURSTS should agree at this point, but if not, select the S2REQSTS, select the CLEAR option and press ENTER.

Map to the Device (See Application User Manual)

The build of the image is complete from the AM side. The next step is to assign the input and/or output signals to the required parameters. This is done through the CLM Configuration Builder and is application-specific. This means that each application will map to the same image point parameters, but the format and conversion process for each specific application will probably vary. Refer to the earlier sections on the CLM Configuration Builder in this manual and the specific application user manual for instructions on mapping your particular device.

Continued on next page

8.2.2.4 Digital Composite Point - 3-State Pulsed, continued

Command the Point

The Digital point may command an output through the S1REQSTS parameter of the image point. *THE DIGITAL POINT MUST HAVE BEEN MAPPED ALREADY THROUGH THE CLM CONFIGURATION BUILDER.*

The steps to command an output are:

- Select the S1REQSTS parameter

- Select the desired state

- Press ENTER

This will PPS the point and cause the CL to execute, sending the appropriate OPn parameter values to the device.

<p>IMPORTANT! A mapped Digital point will not complete the S1REQSTS store if the associated device (NDT) is Out of Service or in the HARD-FAIL condition. This may be checked at the point by examining the DEV_STA parameter. If the value is OFF, then S1REQSTS stores will not complete and the operator will not see the requested change at the Detail Display.</p>

Clear the Point

After certain alarms or other conditions, it may become necessary to clear an existing alarm(s) and/or line up the S1REQSTS and S1CURSTS parameters. This can be done through the second switch on the Digital point. The steps to perform this action are:

- Select the S2REQSTS parameter

- Select the CLEAR state

- Press ENTER

This action will have the impact of setting the S1REQSTS = S1CURSTS, clearing all existing switch alarms, and re-issuing new alarms based on current conditions.

8.2.2.5 Digital Input Point, 3-State

Operation

The 3 State Digital Input point monitors digital devices with a single input. The input state (S1CURSTS) reflects an ON, OFF, (\$ord0-\$ord1), and BAD or COMM_ERR (\$ord2) condition. The D1 parameter reflects the current logical state of the input signal from the Digital device

A CL block is attached to this image point that:

- ⇒ Maps the D1 input signal to one of the ordinal input states (S1CURSTS).
- ⇒ If configured in the alarm options (ALMOPT), issues Off Normal alarms when the input state (D1) changes off of the normal state (PVNORMAL).
- ⇒ If configured in the alarm options (ALMOPT), issues Change-of-State alarms when the input state (D1) changes.

Although this point type is not commandable, the S1REQSTS may be configured to track the current state (S1CURSTS). This is accomplished by use of the second switch. By selecting the S2REQSTS, the switch #1 requested state may be configured to TRACK or NO_TRACK the current state.

Application

Digital device type examples:

- Pump motor monitoring
- Fan motor monitoring
- Digital signals from devices requiring 1 input

Continued on next page

8.2.2.5 Digital Input Point, 3-State, continued

Alarming

The following is a list of alarms that may be issued by this Digital point type. Refer to Table 8-9 for specific alarm definition.

- OUT_SERV
- PT_FAIL
- CFG_ERR_#
- OFF_NORM (If ALMOPT = 1)
- CHGSTATE (If ALMOPT = 2)

Build the Point

The 3-State Digital input point is based on the AM Switch point. There are two methods of building this point type. Both of these methods use the Data Entity Builder (DEB) and the data entry requirements listed in Table 8-13 below.

- 1) Select the AM SWITCH point type and enter data as listed below.

Or

- 2) Copy the DIN3.EB Exception Build file (included and installed with the Platform release media) to another file for editing.

Edit the new Exception Build file using the data listed below as input for each of the points to be included within the file. Cut and Paste as required.

Perform an Exception Build process on the new file.

Refer to Table 8-10 for complete descriptions of all Digital point parameters.

See Figure 8-5 for an Exception Build file example.

Continued on next page

8.2.2.5 Digital Input Point - 3-State, continued

Table 8-13 3-State Digital Input Parameter Entries

Parameter	Entry	Required?	Notes
NAME		Yes	See PKGNAME(3) notes below
UNIT		Yes	
CLSLOTS	1	Yes	
NOPKG	2 or 3	Yes	
NUMSWTCH	2	Yes	
PKGNAME(1)	CLM_DMD3	Yes	
PKGNAME(2)	DIN3CDS	Yes	
PKGNAME(3)	See Notes	-	Application Specific - Refer to the User Manual section for any required packages
S1NSTATE	3	Yes	Enter text for the ordinal 0 (or top) state Enter text for the ordinal 1 (or middle) state Ordinal 2 (or bottom) state text. This state must be the BAD state. Key Access Level for command changes. Normally not used, but may be set to indicate the "normal" state when using off-normal alarming. (s2reqsts = no_track)
S1STATE(0)	See Notes	Yes	
S1STATE(1)	See Notes	Yes	
S1STATE(2)	BAD	Yes	
S1ACCLVL	Operator Supervisor Engineer Program	No	
S2NSTATE	2	Yes	Selected state for "tracking" option Selected state for "No tracking". Key Access Level for setting the tracking option. Default is ENGINEER.
S2STATE(0)	TRACK	Yes	
S2STATE(1)	NO_TRACK	Yes	
S2ACCLVL	Operator Supervisor Engineer Program	No	
PERIOD	NOPERIOD	Yes	CLM Digital points are PPS'd for execution by data received by exception input or the CLM Point Interface Table (PIT) scheduler.
D1_1	\$ord0-\$ord1	Yes	S1CURSTS becomes the ordinal state identified when D1 is ON.
ALMOPT	0-2	No	Alarming Options: 0 = Configuration Alarms Only 1 = Off-Normal Alarming 2 = Chg-State Alarming
PVNORMAL	\$ord0-\$ord1	Yes IF ALMOPT=1	Not required when ALMOPT = 0 or 2. Required when ALMOPT = 1.

Continued on next page

8.2.2.5 Digital Input Point - 3-State, continued

Build Example

The user wants to build a Digital point to monitor the running state of pump 1. The running state is monitored through 1 input signal (RUNNING = ON). The normal state of this pump is to be stopped and the operator needs to be notified by an alarm whenever the pump is running (i.e. off-normal alarming). The interface is Bristol RTU and therefore requires an additional CDS package (refer to the Bristol-Babcock RTU Application User Manual for CLM). Figure 8-5 below shows an example of an Exception Build file (in LCN R500 format) that could be used to create this image point.

Figure 8-5 Build Example for 3-State Digital Input

```
{SCREEN ENTITY PUMP_1( ) }
&T SWITCHAM
&N PUMP_1
UNIT = C1
PTDESC ="PUMP 1 OPERATION "
KEYWORD =" "
CLSLOTS = 1
NOPKG = 3
NUMSWTCH = 2 {**NOTE: Will cause CL LINK ERROR if not set to 2**}
PRIMMOD = _____
SALMDSC1 =" "
SALMDSC2 =" "
SALMDSC3 =" "
BCLEALPR = LOW {**NOTE: R500 And Later**}
BCLFALPR = LOW {**NOTE: R500 And Later**}
CLEALMPR = LOW {**NOTE: R500 And Later**}
CLFALMPR = LOW {**NOTE: R500 And Later**}
CNFERRPR = LOW {**NOTE: R500 And Later**}
SWALM1PR = LOW {**NOTE: R500 And Later**}
SWALM2PR = LOW {**NOTE: R500 And Later**}
SWALM3PR = LOW {**NOTE: R500 And Later**}
ALENBST = ENABLE
PKGNAME(1) ="CLM_DMD3"
PKGNAME(2) ="DIN3CDS "
PKGNAME(3) ="B_DG3CDS" {**NOTE: Provided and required by the Interface**}
S1NSTATE = 3
S1STATES(0) ="RUNNING" {**NOTE: User-entered text**}
S1STATES(1) ="STOPPED" {**NOTE: User-entered text**}
S1STATES(2) ="BAD "
S1ACCLVL =ENGINEER
S1CURSTS = STOPPED
S1REQSTS = STOPPED
```

Continued on next page

8.2.2.5 Digital Input Point - 3-State, continued

Figure 8-5 Build Example for 3-State Digital Input, continued

```
S2NSTATE = 2
S2STATES(0) = "TRACK "
S2STATES(1) = "NO_TRACK"
S2ACCLVL = OPERATOR
S2CURSTS = NO_TRACK
S2REQSTS = NO_TRACK
$IPPASN = OFF
PERIOD = NOPERIOD          {**NOTE: Scheduled and driven by associated PIT**}
CLM_PORT = " "
CLM_DEV = " "
CLM_PIT = " "
D1_1 = $ORD1               {**NOTE: Pump is ON when state is logical one**}
ALMOPT = 1                 {**NOTE: Set to 1 for Off-Normal alarming option**}
PVNORMAL = $ORD1           {**NOTE: STOPPED is the normal state**}
DEV_STA = OFF
PT_FAIL = OFF
```

Continued on next page

8.2.2.5 Digital Input Point - 3-State, continued

Attach the CL Block

All CLM Digital points require a CL Block. The 3-State Digital Input uses the DIN3CL.AO block included with the CLM Platform release media. Use the EC file included to link the block to the 3-State point.

From the COMMAND PROCESSOR window of the Engineering Personality:

```
EC DIN3.EC pppppppp
      where
      pppppppp is the 3-State point name
```

NOTE: 1) Be sure that your pathnames are set for the CL SOURCE/OBJ to point to the proper directory.
2) When executing this EC file for the first time, it is normal to get an error on the UNLINK since the block has not yet been attached.

Link Caution!

If the number of switches (NUMSWTCH) on the image point is not set to 2, the following error will be issued by the Linker.

Error during attempt to get parameter information; PA Status: END OF USER SPEC

Correct and rebuild the image point before attempting to re-link.

Activate the Point

At the Detail Display of the Digital point, set the PTEXECST = ACTIVE. You may need to PROCESS the point again to clear the RESTART state to None.

Map to the Device (See Application User Manual)

The build of the image is complete from the AM side. The next step is to assign the input and/or output signals to the required parameters. This is done through the CLM Configuration Builder and is application-specific. This means that each application will map to the same image point parameters, but the format and conversion process for each specific application will probably vary. Refer to the earlier sections on the CLM Configuration Builder in this manual and the specific application user manual for instructions on mapping your particular device.

Continued on next page

8.2.2.5 Digital Input Point - 3-State, continued

Tracking Option

Since there is no commanded output on the Digital Input point, the command state (S1REQSTS) may be configured to follow, or track, the current input state (S1CURSTS). This option is configured through the second switch (S2REQSTS) on the Digital Input point. Configuration is as follows:

TRACK	S1REQSTS set equal to S1CURSTS on point update.
NO_TRACK	S1REQSTS is not changed on point update. Note that the S1REQSTS may be used to display the normal state in this case if the off-normal alarming is configured (ALMOPT =1).

Off-Normal and Change-of-State alarming are not affected by the tracking option.

8.2.2.6 Digital Output Point, 3-State Latched

Operation

The 3 State Digital Output point emulates digital devices with no inputs and either one or two outputs (e.g. pumps, motors). The output states (S1REQSTS/S1CURSTS) reflect an ON, OFF (\$ord0-\$ord1), and BAD or COMM_ERR (\$ord2) condition. The OP1 and OP2 parameters correspond to the latched output signals required to drive the Digital device, while the FB1 and FB2 parameters monitor the output feedback signals from the device.

A CL block is attached to this image point that:

- ⇒ When changed by the operator, maps the ordinal output state (S1REQSTS) to the OP1 and OP2 output signals, and executes the send to the Digital device.
- ⇒ When an operator change is in progress, monitors the feedback signals (FB1, FB2) for the requested state change against the configured time allowed for the transition (FBTIME), and either alarms or sets the ordinal state (S1CURSTS) to reflect the change.

When an operator commands a change (S1REQSTS), the configured output combination is sent to the Digital device. The output feedback signals (FB1, FB2) are checked to verify that the output signals (OP1, OP2) were properly received and processed by the Digital device. If this state change is not detected before the feedback timer (FBTIME) expires, a CMD_FAIL or CMDDIS alarm is issued.

Alarms may be cleared or the states aligned by selecting the secondary switch (S2REQSTS) option for CLEAR. The CL will set the state back to NOACTION when complete.

Application

Digital device type examples:

- Pump motor control
- Fan motor control
- Devices requiring 1 or 2 latched output signals with feedbacks only

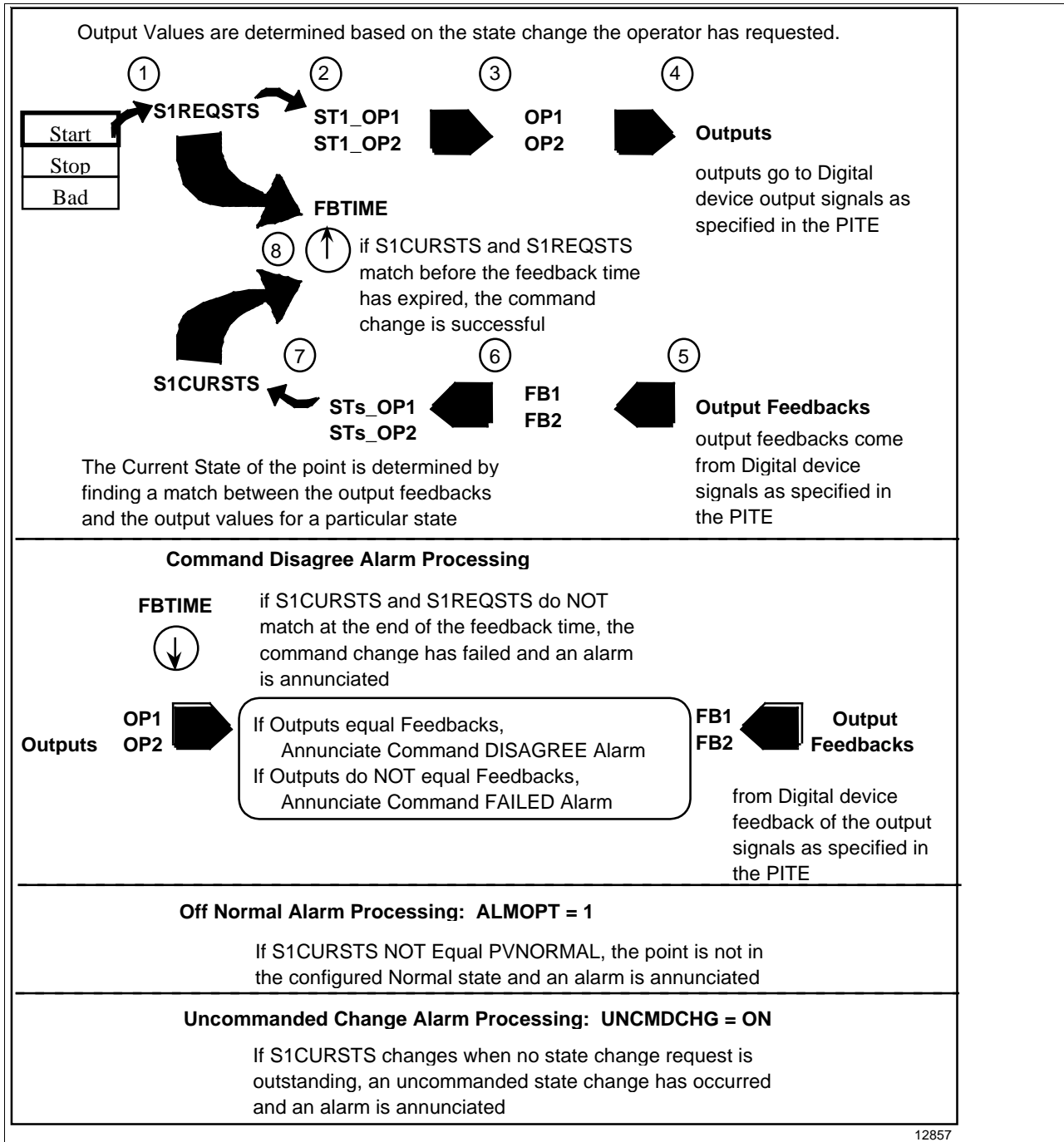
Continued on next page

8.2.2.6 Digital Output Point - 3-State Latched, continued

Operational Flow

Following is a depiction of how a 3-State digital output operates. It shows two output signals that are **Latched** Outputs with no input signals.

Figure 8-6 Operational Flow for 3-State Latched Digital Output



Continued on next page

8.2.2.6 Digital Output Point - 3-State Latched, continued

Alarming

The following is a list of alarms that may be issued by this Digital point type. Refer to Table 8-9 for specific alarm definition.

- OUT_SERV
 - PT_FAIL
 - CFG_ERR_*n*
 - INVCMD
 - CMDFAIL
 - CMDDIS
 - OFF_NORM (If ALMOPT = 1)
 - UNC_CHG (If configured through ALMUNCHG)
 - BAD_PV
 - BADSTATE
-

Build the Point

The 3-State Digital point is based on the AM Switch point. There are two methods of building this point type. Both of these methods use the Data Entity Builder (DEB) and the data entry requirements listed in Table 8-14 below.

- 1) Select the AM SWITCH point type and enter data as listed below.

or

- 2) Copy the DOUT3L.EB Exception Build file (included and installed with the Platform release media) to another file for editing.

Edit the new Exception Build file using the data listed below as input for each of the points to be included within the file. Cut and Paste as required.

Perform an Exception Build process on the new file.

Refer to Table 8-10 for complete descriptions of all Digital point parameters.

See Figure 8-7 for an Exception Build file example.

Continued on next page

8.2.2.6 Digital Output Point - 3-State Latched, continued

Table 8-14 3-State Latched Digital Output Parameter Entries

Parameter	Entry	Required?	Notes
NAME		Yes	See PKGNAME(4) notes below
UNIT		Yes	
CLSLOTS	1	Yes	
NOPKG	3 or 4	Yes	
NUMSWTCH	2	Yes	
PKGNAME(1)	CLM_DMD3	Yes	
PKGNAME(2)	DIG3CDS	Yes	
PKGNAME(3)	DIG3CDS1	Yes	
PKGNAME(4)	See Notes	-	
S1NSTATE	3	Yes	Enter text for the ordinal 0 (or top) state Enter text for the ordinal 1 (or middle) state Ordinal 2 (or bottom) state text. This state must be the BAD state. Key Access Level for command changes. Default is ENGINEER.
S1STATE(0)	See Notes	Yes	
S1STATE(1)	See Notes	Yes	
S1STATE(2)	BAD	Yes	
S1ACCLVL	Operator Supervisor Engineer Program	No	
S2NSTATE	2	Yes	Selected state for “resetting” the point. Default state. Selecting has no impact. Key Access Level for clearing switch alarms and setting commanded state = current state. This setting should match the S1ACCLVL. Default is ENGINEER.
S2STATE(0)	CLEAR	Yes	
S2STATE(1)	NOACTION	Yes	
S2ACCLVL	Operator Supervisor Engineer Program	No	
PERIOD	NOPERIOD	Yes	CLM Digital points are PPS'd for execution by data received by exception input or the CLM Point Interface Table (PIT) scheduler.

Continued on next page

8.2.2.6 Digital Output Point - 3-State Latched, continued

Table 8-14 3-State Latched Digital Output Parameter Entries, continued

Parameter	Entry	Required?	Notes
NUMCMDST	1-2	Yes	Set to 1 if only ordinal state 0 can be commanded. Set to 2 if both ordinal 0 and 1 can be commanded.
ALMOPT	0-1	No	Default = 0 for operational alarms only. Set to 1 to add Off Normal alarming.
ALMUNCHG	ON or OFF	No	Default = OFF. Set to ON to alarm uncommanded state changes.
PVNORMAL	\$ord0-\$ord1	Yes IF ALMOPT=1	Not required when ALMOPT = 0. Required when ALMOPT = 1.
NODINPTS	0	Yes	No Digital Input signals are configured.
FBTIME	See Notes	Yes	Feedback Timeout. Maximum time (in secs) allowed for the commanded state change. Refer to Table 8-10.
TIMEOP	OFF	No	Default = OFF for "Latched" outputs.
NODOPTS	1-2	Yes	Number of Digital output signals.
ST0_OP1	ON or OFF	Yes	For ordinal state 0, bit 1. Set to ON for a logical "1", OFF for a logical "0".
ST0_OP2	ON or OFF	Yes IF NODOPTS=2	For ordinal state 0, bit 2. IF NODOPTS = 2, set to ON for a logical "1", OFF for a logical "0".
ST1_OP1	ON or OFF	Yes	For ordinal state 1, bit 1. Set to ON for a logical "1", OFF for a logical "0".
ST1_OP2	ON or OFF	Yes IF NODOPTS=2	For ordinal state 1, bit 2. IF NODOPTS = 2, set to ON for a logical "1", OFF for a logical "0".
UNCLRTIM	-1, 0, or 1- <i>n</i>	Yes IF ALMUNCHG= ON	Configurable Clear of the Uncommanded Change alarm. IF ALMUNCHG = ON, set to: -1 for Operator clear only, 0 for automatic clear on next point execution 1- <i>n</i> for automatic clear after <i>n</i> seconds.

Continued on next page

8.2.2.6 Digital Output Point - 3-State Latched, continued

Build Example

A Digital point needs to be built to start and stop pump #1. The pump requires output signal #1 = ON for RUN and output signal #2 = ON for OFF. There are no input signals to monitor the running state, so the output feedbacks must be checked to be sure that the command was accepted by the Digital device. The response time from the Digital device to set the commanded state is approximately 10 seconds. The operator needs to be able to command the pump and wants to know if the pump is started or stopped at the pump (uncommanded change). The interface is Modbus and therefore requires no additional CDS packages. Figure 8-7 below shows an example of an Exception Build file (in LCN R500 format) that could be used to create this image point.

Figure 8-7 Build Example for 3-State Latched Digital Output

```
{SCREEN ENTITY PUMP_1() }
&T SWITCHAM
&N PUMP_1
UNIT = C1
PTDESC ="PUMP 1 CONTROL"
KEYWORD =" "
CLSLOTS = 1
NOPKG = 3
NUMSWTCH = 2          {**NOTE: Will cause CL LINK ERROR if not set to 2**}
PRIMMOD = _____
SALMDSC1 =" "
SALMDSC2 =" "
SALMDSC3 =" "
BCLEALPR = LOW        {**NOTE: R500 And Later**}
BCLFALPR = LOW        {**NOTE: R500 And Later**}
CLEALMPR = LOW        {**NOTE: R500 And Later**}
CLFALMPR = LOW        {**NOTE: R500 And Later**}
CNFERRPR = LOW        {**NOTE: R500 And Later**}
SWALM1PR = LOW        {**NOTE: R500 And Later**}
SWALM2PR = LOW        {**NOTE: R500 And Later**}
SWALM3PR = LOW        {**NOTE: R500 And Later**}
ALENBST = ENABLE
PKGNAME(1) ="CLM_DMD3"
PKGNAME(2) ="DIG3CDS "
PKGNAME(3) ="DIG3CDS1"
SINSTATE = 3
S1STATES(0) ="START " {**NOTE: User-entered text**}
S1STATES(1) ="STOP  " {**NOTE: User-entered text**}
S1STATES(2) ="BAD   "
S1ACCLVL = OPERATOR
S1CURSTS = STOP
S1REQSTS = STOP
```

Continued on next page

8.2.2.6 Digital Output Point - 3-State Latched, continued

Figure 8-7 Build Example for 3-State Latched Digital Output , continued

```
S2NSTATE = 2
S2STATES(0) ="CLEAR  "
S2STATES(1) ="NOACTION"
S2ACCLVL = OPERATOR
S2CURSTS = NOACTION
S2REQSTS = NOACTION
$IPPASN = OFF
PERIOD = NOPERIOD          {**NOTE: Scheduled and driven by associated PIT**}
CLM_PORT =" "
CLM_DEV =" "
CLM_PIT =" "
NODINPTS = 0
NUMCMDST = 2
ALMOPT = 0
ALMUNCHG = ON
PVNORMAL = $ORD1
FBTIME = 20.0
TIMEOP = OFF
NODOPTS = 2
ST0_OP1 = ON               {**NOTE: START state sets output signal #1 to ON**}
ST0_OP2 = OFF
ST1_OP1 = OFF
ST1_OP2 = ON               {**NOTE: STOP state sets output signal #2 to ON**}
UNC_OP1 = OFF
UNC_OP2 = OFF
UNCLRTIM = -1
UNCCLEAR = OFF
DEV_STA = OFF
PT_FAIL = OFF
CMDCHG = OFF
ALMTYPE1 = A_NONE
ALMTYPE2 = A_NONE
ALMTYPE3 = A_NONE
INITLIZD = OFF
TIMER = 0
DATA(1) = 0
DATA(2) = 0
DATA(3) = 0
DATA(4) = 0
DATA(5) = 0
DATA(6) = 0
DATA(7) = 0
DATA(8) = 0
DATA(9) = 0
DATA(10) = 0
```

Continued on next page

8.2.2.6 Digital Output Point - 3-State Latched, continued

Attach the CL Block

All CLM Digital points require a CL Block. The 3-State Digital Output uses the DIG3CL.AO block included with the CLM Platform release media. Use the EC file included to link the block to the 3-State point.

From the COMMAND PROCESSOR window of the Engineering Personality:

```
EC DIG3.EC pppppppp
where
pppppppp is the 3-State point name
```

NOTE: 1) Be sure that your pathnames are set for the CL SOURCE/OBJ to point to the proper directory.
2) When executing this EC file for the first time, it is normal to get an error on the UNLINK since the block has not yet been attached.

Link Caution!

If the number of switches (NUMSWTCH) on the image point is not set to 2, the following error will be issued by the Linker.

Error during attempt to get parameter information; PA Status: END OF USER SPEC

Correct and rebuild the image point before attempting to re-link.

Activate the Point

At the Detail Display of the Digital point, set the PTEXECST = ACTIVE. You may need to PROCESS the point again to clear the RESTART state to NONE. The S1REQSTS and S1CURSTS should agree at this point, but if not, select the S2REQSTS, select the CLEAR option and press ENTER.

Map to the Device (See Application User Manual)

The build of the image is complete from the AM side. The next step is to assign the input and/or output signals to the required parameters. This is done through the CLM Configuration Builder and is application-specific. This means that each application will map to the same image point parameters, but the format and conversion process for each specific application will probably vary. Refer to the earlier sections on the CLM Configuration Builder in this manual and the specific application user manual for instructions on mapping your particular device.

Continued on next page

8.2.2.6 Digital Output Point - 3-State Latched, continued

Command the Point

The Digital point may command an output through the S1REQSTS parameter of the image point. *THE DIGITAL POINT MUST HAVE BEEN MAPPED ALREADY THROUGH THE CLM CONFIGURATION BUILDER.*

The steps to command an output are:

- Select the S1REQSTS parameter

- Select the desired state

- Press ENTER

This will PPS the point and cause the CL to execute, sending the appropriate OPn parameter values to the device.

<p>IMPORTANT! A mapped Digital point will not complete the S1REQSTS store if the associated device (NDT) is Out of Service or in the HARD-FAIL condition. This may be checked at the point by examining the DEV_STA parameter. If the value is OFF, then S1REQSTS stores will not complete and the operator will not see the requested change at the Detail Display.</p>

Clear the Point

After certain alarms or other conditions, it may become necessary to clear an existing alarm(s) and/or line up the S1REQSTS and S1CURSTS parameters. This can be done through the second switch on the Digital point. The steps to perform this action are:

- Select the S2REQSTS parameter

- Select the CLEAR state

- Press ENTER

This action will have the impact of setting the S1REQSTS = S1CURSTS, clearing all existing switch alarms, and re-issuing new alarms based on current conditions.

8.2.2.7 Digital Output Point, 3-State Pulsed

Operation

The 3 State Digital Output point emulates digital devices with no inputs and either one or two outputs (e.g. pumps, motors). The output states (S1REQSTS/S1CURSTS) reflect an ON, OFF (\$ord0-\$ord1), and BAD or COMM_ERR (\$ord2) condition. The OP1 and OP2 parameters correspond to the pulsed output signals required to drive the Digital device, while the FB1 and FB2 parameters monitor the output feedback signals from the device.

A CL block is attached to this image point that:

- ⇒ When changed by the operator, maps the ordinal output state (S1REQSTS) to the OP1 and OP2 output signals, and executes the send to the Digital device.
- ⇒ When an operator change is in progress, monitors the feedback signals (FB1, FB2) for the requested state change against the configured time allowed for the transition (FBTIME), and either alarms or sets the ordinal state (S1CURSTS) to reflect the change.

When an operator commands a change (S1REQSTS), the configured output combination is sent to the Digital device. The output feedback signals (FB1, FB2) are checked against the uncommanded output signals (stable state). If the uncommanded state is not detected before the feedback timer (FBTIME) expires, a CMD_FAIL or CMDDIS alarm is issued.

Alarms may be cleared or the states aligned by selecting the secondary switch (S2REQSTS) option for CLEAR. The CL will set the state back to NOACTION when complete.

Application

Digital device type examples:

- Pump motor control
- Fan motor control
- Devices requiring 1 or 2 pulsed output signals with feedbacks only

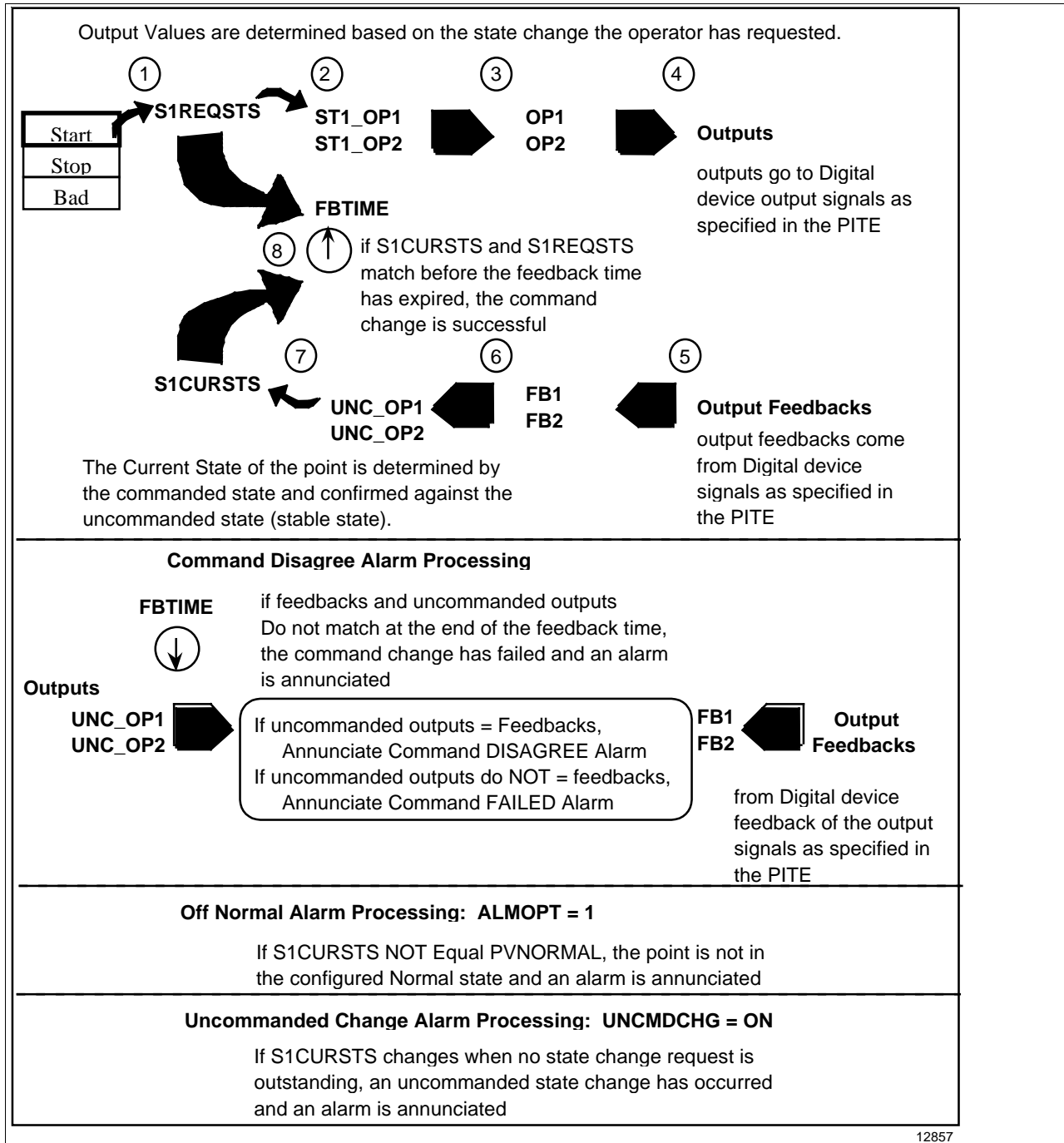
Continued on next page

8.2.2.7 Digital Output Point - 3-State Pulsed, continued

Operational Flow

Following is a depiction of how a 3-State Digital Output operates. It shows two output signals that are **Pulsed** Outputs with no input signals.

Figure 8-8 Operational Flow for 3-State Pulsed Digital Output



Continued on next page

8.2.2.7 Digital Output Point - 3-State Pulsed, continued

Alarming

The following is a list of alarms that may be issued by this Digital point type. Refer to Table 8-9 for specific alarm definition.

- OUT_SERV
 - PT_FAIL
 - CFG_ERR_*n*
 - INVCMD
 - CMDFAIL
 - CMDDIS
 - OFF_NORM (If ALMOPT = 1)
 - UNC_CHG (If configured through ALMUNCHG)
 - BAD_PV
 - BADSTATE
-

Build the Point

The 3-State Digital point is based on the AM Switch point. There are two methods of building this point type. Both of these methods use the Data Entity Builder (DEB) and the data entry requirements listed in Table 8-15 below.

- 1) Select the AM SWITCH point type and enter data as listed below.

Or

- 2) Copy the DOUT3P.EB Exception Build file (included and installed with the Platform release media) to another file for editing.

Edit the new Exception Build file using the data listed below as input for each of the points to be included within the file. Cut and Paste as required.

Perform an Exception Build process on the new file.

Refer to Table 8-10 for complete descriptions of all Digital point parameters.

See Figure 8-9 for an Exception Build file example.

Continued on next page

8.2.2.7 Digital Output Point - 3-State Pulsed, continued

Table 8-15 3-State Pulsed Digital Output Parameter Entries

Parameter	Entry	Required?	Notes
NAME		Yes	See PKGNAME(4) notes below Application Specific - Refer to the User Manual section for any required packages
UNIT		Yes	
CLSLOTS	1	Yes	
NOPKG	3 or 4	Yes	
NUMSWTCH	2	Yes	
PKGNAME(1)	CLM_DMD3	Yes	
PKGNAME(2)	DIG3CDS	Yes	
PKGNAME(3)	DIG3CDS1	Yes	
PKGNAME(4)	See Notes	-	
S1NSTATE	3	Yes	Enter text for the ordinal 0 (or top) state Enter text for the ordinal 1 (or middle) state Ordinal 2 (or bottom) state text. This state must be the BAD state. Key Access Level for command changes. Default is ENGINEER.
S1STATE(0)	See Notes	Yes	
S1STATE(1)	See Notes	Yes	
S1STATE(2)	BAD	Yes	
S1ACCLVL	Operator Supervisor Engineer Program	No	
S2NSTATE	2	Yes	Selected state for "resetting" the point. Default state. Selecting has no impact. Key Access Level for clearing switch alarms and setting commanded state = current state. This setting should match the S1ACCLVL. Default is ENGINEER.
S2STATE(0)	CLEAR	Yes	
S2STATE(1)	NOACTION	Yes	
S2ACCLVL	Operator Supervisor Engineer Program	No	
PERIOD	NOPERIOD	Yes	CLM Digital points are PPS'd for execution by data received by exception input or the CLM Point Interface Table (PIT) scheduler.

Continued on next page

8.2.2.7 Digital Output Point - 3-State Pulsed, continued

Table 8-15 3-State Pulsed Digital Output Parameter Entries, continued

Parameter	Entry	Required?	Notes
NUMCMDST	1-2	Yes	Set to 1 if only ordinal state 0 can be commanded. Set to 2 if both ordinal 0 and 1 can be commanded.
ALMOPT	0-1	No	Default = 0 for operational alarms only. Set to 1 to add Off Normal alarming.
ALMUNCHG	ON or OFF	No	Default = OFF. Set to ON to alarm uncommanded state changes.
PVNORMAL	\$ord0-\$ord1	Yes IF ALMOPT=1	Not required when ALMOPT = 0. Required when ALMOPT = 1.
NODINPTS	0	Yes	No Digital Input signals are configured.
FBTIME	See Notes	Yes	Feedback Timeout. Maximum time (in secs) allowed for the commanded state change. Refer to Table 8-10.
TIMEOP	ON	Yes	Set to ON for "Pulsed" outputs.
NODOPTS	1-2	Yes	Number of Digital output signals.
ST0_OP1	ON or OFF	Yes	For ordinal state 0, bit 1. Set to ON for a logical "1", OFF for a logical "0".
ST0_OP2	ON or OFF	Yes IF NODOPTS=2	For ordinal state 0, bit 2. IF NODOPTS = 2, set to ON for a logical "1", OFF for a logical "0".
UNC_OP1	ON or OFF	Yes	Uncommanded "stable" state for output bit 1. Set to ON for a logical "1", OFF for a logical "0".
UNC_OP2	ON or OFF	Yes IF NODOPTS=2	Uncommanded "stable" state for output bit 2. Set to ON for a logical "1", OFF for a logical "0".
UNCLRTIM	-1, 0, or 1- <i>n</i>	Yes IF ALMUNCHG= ON	Configurable Clear of the Uncommanded Change alarm. IF ALMUNCHG = ON, set to: -1 for Operator clear only, 0 for automatic clear on next point execution 1- <i>n</i> for automatic clear after <i>n</i> seconds.

Continued on next page

8.2.2.7 Digital Output Point - 3-State Pulsed, continued

Build Example

A Digital point needs to be built to start and stop pump #1. The pump requires output signal #1 = ON for RUN and output signal #2 = ON for OFF. There are no input signals to monitor the running state, so the output feedbacks must be checked to be sure that the command was accepted by the Digital device. The Digital device uses pulsed signals with a 4 second duration that returns back to the OFF state in all cases. The operator needs to be able to command the pump and wants to know if the pump is started or stopped at the pump (uncommanded change). The interface is Modbus and therefore requires no additional CDS packages. Figure 8-9 below shows an example of an Exception Build file (in LCN R500 format) that could be used to create this image point.

Continued on next page

8.2.2.7 Digital Output Point - 3-State Pulsed, continued

Figure 8-9 Build Example for 3-State Pulsed Digital Output

```
{SCREEN ENTITY PUMP_1() }
&T SWITCHAM
&N PUMP_1
UNIT = C1
PTDESC ="PUMP 1 CONTROL  "
KEYWORD ="  "
CLSLOTS = 1
NOPKG = 3
NUMSWTCH = 2          {**NOTE: Will cause CL LINK ERROR if not set to 2**}
PRIMMOD = _____
SALMDSC1 ="  "
SALMDSC2 ="  "
SALMDSC3 ="  "
BCLEALPR = LOW        {**NOTE: R500 And Later**}
BCLFALPR = LOW        {**NOTE: R500 And Later**}
CLEALMPR = LOW        {**NOTE: R500 And Later**}
CLFALMPR = LOW        {**NOTE: R500 And Later**}
CNFERRPR = LOW        {**NOTE: R500 And Later**}
SWALM1PR = LOW        {**NOTE: R500 And Later**}
SWALM2PR = LOW        {**NOTE: R500 And Later**}
SWALM3PR = LOW        {**NOTE: R500 And Later**}
ALENBST = ENABLE
PKGNAME(1) ="CLM_DMD3"
PKGNAME(2) ="DIG3CDS "
PKGNAME(3) ="DIG3CDS1"
S1NSTATE = 3
S1STATES(0) ="START  " {**NOTE: User-entered text**}
S1STATES(1) ="STOP  " {**NOTE: User-entered text**}
S1STATES(2) ="BAD   "
S1ACCLVL = OPERATOR
S1CURSTS = STOP
S1REQSTS = STOP
S2NSTATE = 2
```

Continued on next page

8.2.2.7 Digital Output Point - 3-State Pulsed, continued

Figure 8-9 Build Example for 3-State Pulsed Digital Output, continued

```
S2STATES(0) ="CLEAR "
S2STATES(1) ="NOACTION"
S2ACCLVL = OPERATOR
S2CURSTS = NOACTION
S2REQSTS = NOACTION
$IPPASN = OFF
PERIOD = NOPERIOD          {**NOTE: Scheduled and driven by associated PIT**}
CLM_PORT =" "
CLM_DEV =" "
CLM_PIT =" "
NODINPTS = 0
NUMCMDST = 2
ALMOPT = 0
ALMUNCHG = ON
PVNORMAL = $ORD1
FBTIME = 8.0
TIMEOP = ON
NODOPTS = 2
ST0_OP1 = ON               {**NOTE: START state sets output signal #1 to ON**}
ST0_OP2 = OFF
ST1_OP1 = OFF
ST1_OP2 = ON              {**NOTE: STOP state sets output signal #2 to ON**}
UNC_OP1 = OFF
UNC_OP2 = OFF
UNCLRTIM = -1
UNCCLEAR = OFF
DEV_STA = OFF
PT_FAIL = OFF
CMDCHG = OFF
ALMTYPE1 = A_NONE
ALMTYPE2 = A_NONE
ALMTYPE3 = A_NONE
INITLIZD = OFF
TIMER = 0
DATA(1) = 0
DATA(2) = 0
DATA(3) = 0
DATA(4) = 0
DATA(5) = 0
DATA(6) = 0
DATA(7) = 0
DATA(8) = 0
DATA(9) = 0
DATA(10) = 0
```

Continued on next page

8.2.2.7 Digital Output Point - 3-State Pulsed, continued

Attach the CL Block

All CLM Digital points require a CL Block. The 3-State Digital Output uses the DIG3CL.AO block included with the CLM Platform release media. Use the EC file included to link the block to the 3-State point.

From the COMMAND PROCESSOR window of the Engineering Personality:

```
EC DIG3.EC pppppppp
where
pppppppp is the 3-State point name
```

NOTE: 1) Be sure that your pathnames are set for the CL SOURCE/OBJ to point to the proper directory.
2) When executing this EC file for the first time, it is normal to get an error on the UNLINK since the block has not yet been attached.

Link Caution!

If the number of switches (NUMSWTCH) on the image point is not set to 2, the following error will be issued by the Linker.

Error during attempt to get parameter information; PA Status: END OF USER SPEC

Correct and rebuild the image point before attempting to re-link.

Activate the Point

At the Detail Display of the Digital point, set the PTEXECST = ACTIVE. You may need to PROCESS the point again to clear the RESTART state to NONE. The S1REQSTS and S1CURSTS should agree at this point, but if not, select the S2REQSTS, select the CLEAR option and press ENTER.

Map to the Device (See Application User Manual)

The build of the image is complete from the AM side. The next step is to assign the input and/or output signals to the required parameters. This is done through the CLM Configuration Builder and is application-specific. This means that each application will map to the same image point parameters, but the format and conversion process for each specific application will probably vary. Refer to the earlier sections on the CLM Configuration Builder in this manual and the specific application user manual for instructions on mapping your particular device.

Continued on next page

8.2.2.7 Digital Output Point - 3-State Pulsed, continued

Command the Point

The Digital point may command an output through the S1REQSTS parameter of the image point. *THE DIGITAL POINT MUST HAVE BEEN MAPPED ALREADY THROUGH THE CLM CONFIGURATION BUILDER.*

The steps to command an output are:

- Select the S1REQSTS parameter

- Select the desired state

- Press ENTER

This will PPS the point and cause the CL to execute, sending the appropriate OPn parameter values to the device.

<p>IMPORTANT! A mapped Digital point will not complete the S1REQSTS store if the associated device (NDT) is Out of Service or in the HARD-FAIL condition. This may be checked at the point by examining the DEV_STA parameter. If the value is OFF, then S1REQSTS stores will not complete and the operator will not see the requested change at the Detail Display.</p>

Clear the Point

After certain alarms or other conditions, it may become necessary to clear an existing alarm(s) and/or line up the S1REQSTS and S1CURSTS parameters. This can be done through the second switch on the Digital point. The steps to perform this action are:

- Select the S2REQSTS parameter

- Select the CLEAR state

- Press ENTER

This action will have the impact of setting the S1REQSTS = S1CURSTS, clearing all existing switch alarms, and re-issuing new alarms based on current conditions.

8.2.2.8 Digital Composite Point, 4-State Latched

Operation

The 4 State Digital Composite point emulates digital devices with two inputs and either one or two outputs (e.g. valves). The input states (S1CURSTS) reflect an OPEN, CLOSE, INBTWEEN (\$ord0-\$ord2), and BAD or COMM_ERR (\$ord3) condition. The D1 and D2 parameters reflect the current state of the Digital device, while the OP1, OP2 and OP3 parameters correspond to the latched output signals required to drive the Digital device.

NOTE: Although 3 output signals are allowed for the 4-State composite, only 2 states are commandable. This means it takes 3 output signals to map the OPEN state and 3 output signals to map the CLOSE state. The INBTWEEN state cannot be commanded from this point. See the 5-state point definition.

A CL block is attached to this image point that:

- ⇒ Maps the D1 and D2 input signals to one of the ordinal input states (S1CURSTS).
- ⇒ When changed by the operator, maps the ordinal output state (S1REQSTS) to the OP1, OP2 and OP3 output signals, and executes the send to the Digital device.
- ⇒ When an operator change is in progress, monitors the input signals (D1 and D2) for the requested state change against the configured time allowed for the transition (FBTIME), and either alarms or sets the ordinal state (S1CURSTS) to reflect the change.
- ⇒ If configured in ALMUNCHG, alarms when an input state change (D1 and D2) occurs that was not commanded by the operator.
- ⇒ If configured in the alarm options (ALMOPT), issues Off Normal alarms when the input state (D1 and D2) changes off of the normal state (PVNORMAL).

When an operator commands a change (S1REQSTS), the configured output combination is sent to the Digital device. The output feedback signals (FB1, FB2, FB3) are checked to verify that the output signals (OP1, OP2, OP3) were properly received and processed by the Digital device. The input signals (D1, D2) are then monitored for the commanded state change. If this state change is not detected before the feedback timer (FBTIME) expires, a CMD_FAIL or CMDDIS alarm is issued.

Alarms may be cleared or the states aligned by selecting the secondary switch (S2REQSTS) option for CLEAR. The CL will set the state back to NOACTION when complete.

Application

Digital device type examples:

Pump motor monitor and control

Valve control

Devices requiring 2 inputs and 1 to 3 latched output signals

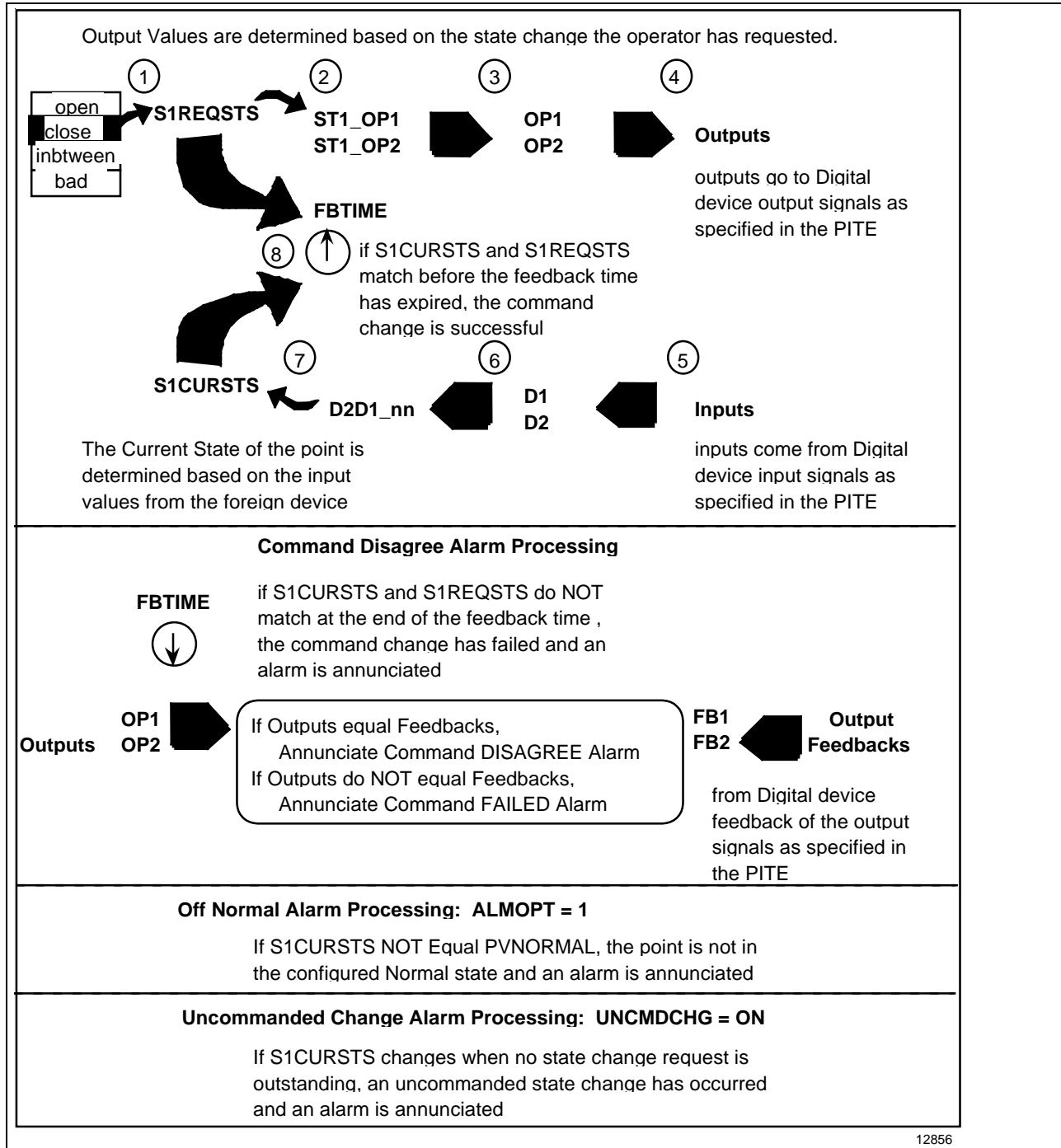
Continued on next page

8.2.2.8 Digital Composite Point - 4-State Latched, continued

Operational Flow

Following is a depiction of how a 4-State Digital Composite operates. It shows a Digital Composite with two output signals that are **Latched** Outputs with two input signals.

Figure 8-10 Operational Flow for 4-State Latched Digital Composite



Continued on next page

8.2.2.8 Digital Composite Point - 4-State Latched, continued

I/O Configuration Options

Number of Inputs	Number of Outputs	Max Cmd States
2	1	2
2	2	2
2	3	2

Alarming

The following is a list of alarms that may be issued by this Digital point type. Refer to Table 8-9 for specific alarm definition.

- OUT_SERV
- PT_FAIL
- CFG_ERR_*n*
- INVCMD
- CMDFAIL
- CMDDIS
- NO_MOVE
- UNC_CHG (If configured through ALMUNCHG)
- OFF_NORM (If ALMOPT = 1)
- BAD_PV
- BADSTATE

Build the Point

The 4-State Digital point is based on the AM Switch point. There are two methods of building this point type. Both of these methods use the Data Entity Builder (DEB) and the data entry requirements listed in Table 8-16 below.

- 1) Select the AM SWITCH point type and enter data as listed below.
or
- 2) Copy the DIG4L.EB Exception Build file (included and installed with the Platform release media) to another file for editing.

Edit the new Exception Build file using the data listed below as input for each of the points to be included within the file. Cut and Paste as required.

Perform an Exception Build process on the new file.

Refer to Table 8-10 for complete descriptions of all Digital point parameters.

See Figure 8-11 for an Exception Build file example.

Continued on next page

8.2.2.8 Digital Composite Point - 4-State Latched, continued

Table 8-16 4-State Latched Digital Point Parameter Entries

Parameter	Entry	Required?	Notes
NAME		Yes	See PKGNAME(4) notes below Application Specific - Refer to the User Manual section for any required packages
UNIT		Yes	
CLSLOTS	1	Yes	
NOPKG	3 or 4	Yes	
NUMSWTCH	2	Yes	
PKGNAME(1)	CLM_DMD3	Yes	
PKGNAME(2)	DIG4CDS	Yes	
PKGNAME(3)	DIG4CDS1	Yes	
PKGNAME(4)	See Notes	-	
S1NSTATE	4	Yes	Enter text for the ordinal 0 (or top) state Enter text for the ordinal 1 state Enter text for the ordinal 2 (between or moving) state Ordinal 3 (or bottom) state text. This state must be the BAD state. Key Access Level for command changes. Default is ENGINEER.
S1STATE(0)	See Notes	Yes	
S1STATE(1)	See Notes	Yes	
S1STATE(2)	See Notes	Yes	
S1STATE(3)	BAD	Yes	
S1ACCLVL	Operator Supervisor Engineer Program	No	
S2NSTATE	2	Yes	Selected state for "resetting" the point. Default state. Selecting has no impact. Key Access Level for clearing switch alarms and setting commanded state = current state. This setting should match the S1ACCLVL. Default is ENGINEER.
S2STATE(0)	CLEAR	Yes	
S2STATE(1)	NOACTION	Yes	
S2ACCLVL	Operator Supervisor Engineer Program	No	
PERIOD	NOPERIOD	Yes	CLM Digital points are PPS'd for execution by data received by exception input or the CLM Point Interface Table (PIT) scheduler.

Continued on next page

8.2.2.8 Digital Composite Point - 4-State Latched, continued

Table 8-16 4-State Latched Digital Point Parameter Requirements, continued

Parameter	Entry	Required?	Notes
D1_0	\$ord0-\$ord1	Yes	S1CURSTS becomes the ordinal state identified when D1 is OFF.
D1_1	\$ord0-\$ord1	Yes	S1CURSTS becomes the ordinal state identified when D1 is ON.
D2D1_01	\$ord0-\$ord3	Yes	Enter the ordinal state that maps to the condition where bit 1 (D1)=ON and bit 2 (D2)=OFF
D2D1_10	\$ord0-\$ord3	Yes	Enter the ordinal state that maps to the condition where bit 1 (D1)=OFF and bit 2 (D2)=ON
D2D1_11	\$ord0-\$ord3	Yes	Enter the ordinal state that maps to the condition where bit 1 (D1)=ON and bit 2 (D2)=ON
D2D1_00	\$ord0-\$ord3	Yes	Enter the ordinal state that maps to the condition where bit 1 (D1)=OFF and bit 2 (D2)=OFF
NUMCMDST	1-2	Yes	Set to 1 if only ordinal state 0 can be commanded. Set to 2 if both ordinal 0 and 1 can be commanded.
ALMOPT	0-1	No	Default = 0 for operational alarms only. Set to 1 to add Off Normal alarming.
ALMUNCHG	ON or OFF	No	Default = OFF. Set to ON to alarm uncommanded state changes.
PVNORMAL	\$ord0-\$ord1	Yes IF ALMOPT=1	Not required when ALMOPT = 0. Required when ALMOPT = 1.
NODINPTS	2	Yes	Number of Digital Input signals is 2.
FBTIME	See Notes TT.tt	Yes	Feedback Timeouts. Maximum times (in secs) allowed for the commanded state changes, where TT=total time, tt=no movement time. Refer to Table 8-10.
TIMEOP	OFF	Yes	Set to OFF for "Latched" outputs.
NODOPTS	1-3	Yes	Number of Digital output signals.
ST0_OP1	ON or OFF	Yes	For ordinal state 0, bit 1. Set to ON for a logical "1", OFF for a logical "0".
ST0_OP2	ON or OFF	Yes IF NODOPTS=2	For ordinal state 0, bit 2. IF NODOPTS = 2, set to ON for a logical "1", OFF for a logical "0".
ST0_OP3	ON or OFF	Yes IF NODOPTS=2	For ordinal state 0, bit 3. IF NODOPTS = 3, set to ON for a logical "1", OFF for a logical "0".

Continued on next page

8.2.2.8 Digital Composite Point - 4-State Latched, continued

Table 8-16 4-State Latched Digital Point Parameter Requirements, continued

Parameter	Entry	Required?	Notes
ST1_OP1	ON or OFF	Yes	For ordinal state 1, bit 1. Set to ON for a logical "1", OFF for a logical "0".
ST1_OP2	ON or OFF	Yes IF NODOPTS=2	For ordinal state 1, bit 2. IF NODOPTS = 2, set to ON for a logical "1", OFF for a logical "0".
ST1_OP3	ON or OFF	Yes IF NODOPTS=2	For ordinal state 1, bit 3. IF NODOPTS = 3, set to ON for a logical "1", OFF for a logical "0".
UNCLRTIM	-1, 0, or 1- <i>n</i>	Yes IF ALMUNCHG= ON	Configurable Clear of the Uncommanded Change alarm. IF ALMUNCHG = ON, set to: -1 for Operator clear only, 0 for automatic clear on next point execution 1- <i>n</i> for automatic clear after <i>n</i> seconds.

Continued on next page

8.2.2.8 Digital Composite Point - 4-State Latched, continued

Build Example

The user wants to build a Digital point to emulate a valve. The valve requires output signal #1 = ON for CLOSE and output signal #2 = ON for OPEN. The running state of the valve is monitored through the two limit switches. Input signal #1 (D1) from the CLOSED limit switch and input signal #2 (D2) from the OPENED limit switch. The limit switches are normally open, so the input signals reflect OFF and OFF when the valve is MOVING or inbetween stops. It takes approximately 45 seconds for the valve to completely transition from one stop to the other. If the valve has not started moving after 15 seconds, then there is a problem or a jam. The operator needs to be able to command the valve and it cannot be started or stopped at the valve (no uncommanded change). The interface is through a Modbus interface to a PLC and therefore no additional CDS packages are required. Figure 8-11 below shows an example of an Exception Build file (in LCN R500 format) that could be used to create this image point.

Figure 8-11 Build Example for 4-State Latched Digital Composite

```
{SCREEN ENTITY VALVE_1( ) }
&T SWITCHAM
&N VALVE_1
UNIT = C1
PTDESC = "PUMP 1 CONTROL"
KEYWORD = " "
CLSLOTS = 1
NOPKG = 3
NUMSWTCH = 2 {**NOTE: Will cause CL LINK ERROR if not set to 2**}
PRIMMOD = _____
SALMDSC1 = " "
SALMDSC2 = " "
SALMDSC3 = " "
BCLEALPR = LOW {**NOTE: R500 And Later**}
BCLFALPR = LOW {**NOTE: R500 And Later**}
CLEALMPR = LOW {**NOTE: R500 And Later**}
CLFALMPR = LOW {**NOTE: R500 And Later**}
CNFERRPR = LOW {**NOTE: R500 And Later**}
SWALM1PR = LOW {**NOTE: R500 And Later**}
SWALM2PR = LOW {**NOTE: R500 And Later**}
SWALM3PR = LOW {**NOTE: R500 And Later**}
ALENBST = ENABLE
PKGNAME(1) = "CLM_DMD3"
PKGNAME(2) = "DIG4CDS "
PKGNAME(3) = "DIG4CDS1"
```

Continued on next page

8.2.2.8 Digital Composite Point - 4-State Latched, continued

Figure 8-11 Build Example for 4-State Latched Digital Composite , continued

```
S1NSTATE = 4
S1STATES(0) = "OPEN " {**NOTE: User-entered text**}
S1STATES(1) = "CLOSE" {**NOTE: User-entered text**}
S1STATES(2) = "MOVING "
S1STATES(3) = "BAD "
S1ACCLVL = OPERATOR
S1CURSTS = CLOSE
S1REQSTS = CLOSE
S2NSTATE = 2
S2STATES(0) = "CLEAR "
S2STATES(1) = "NOACTION"
S2ACCLVL = OPERATOR
S2CURSTS = NOACTION
S2REQSTS = NOACTION
$IPASN = OFF
PERIOD = NOPERIOD {**NOTE: Scheduled and driven by associated PIT**}
CLM_PORT = " "
CLM_DEV = " "
CLM_PIT = " "
NODINPTS = 2
D1_0 = $ORD0 {**NOTE: Valve is OPEN when state is logical zero**}
D1_1 = $ORD1
D2D1_01 = $ORD1 {**CLOSE**}
D2D1_10 = $ORD0 {**OPEN**}
D2D1_00 = $ORD2 {**MOVING**}
D2D1_11 = $ORD3 {**BAD**}
NUMCMDST = 2
ALMOPT = 0
ALMUNCHG = OFF
PVNORMAL = $ORD1
FBTIME = 45.15
TIMEOP = OFF {**NOTE: MUST be OFF for Latched output signals**}
NODOPTS = 2
ST0_OP1 = OFF {**NOTE: OPEN state sets output signal #2 to ON**}
ST0_OP2 = ON
ST0_OP3 = OFF
ST1_OP1 = ON {**NOTE: CLOSE state sets output signal #1 to ON**}
ST1_OP2 = OFF
ST1_OP3 = OFF
UNCLRTIM = -1
UNCCLEAR = OFF
DEV_STA = OFF
PT_FAIL = OFF
CMDCHG = OFF
```

Continued on next page

8.2.2.8 Digital Composite Point - 4-State Latched, continued

Figure 8-11 Build Example for 4-State Latched Digital Composite, continued

```
ALMTYPE1 = A_NONE
ALMTYPE2 = A_NONE
ALMTYPE3 = A_NONE
INITLIZD = OFF
TIMER   = 0
DATA(1) = 0
DATA(2) = 0
DATA(3) = 0
DATA(4) = 0
DATA(5) = 0
DATA(6) = 0
DATA(7) = 0
DATA(8) = 0
DATA(9) = 0
DATA(10) = 0
```

Continued on next page

8.2.2.8 Digital Composite Point - 4-State Latched, continued

Attach the CL Block

All CLM Digital points require a CL Block. The 4-State Digital Composite uses the DIG4CL.AO block included with the CLM Platform release media. Use the EC file included to link the block to the 4-State point.

From the COMMAND PROCESSOR window of the Engineering Personality:

```
EC DIG4.EC pppppppp
      where
      pppppppp is the 4-State point name
```

NOTE: 1) Be sure that your pathnames are set for the CL SOURCE/OBJ to point to the proper directory.
2) When executing this EC file for the first time, it is normal to get an error on the UNLINK since the block has not yet been attached.

Link Caution!

If the number of switches (NUMSWTCH) on the image point is not set to 2, the following error will be issued by the Linker.

Error during attempt to get parameter information; PA Status: END OF USER SPEC

Correct and rebuild the image point before attempting to re-link.

Activate the Point

At the Detail Display of the Digital point, set the PTEXECST = ACTIVE. You may need to PROCESS the point again to clear the RESTART state to None. The S1REQSTS and S1CURSTS should agree at this point, but if not, select the S2REQSTS, select the CLEAR option and press ENTER.

Map to the Device (See Application User Manual)

The build of the image is complete from the AM side. The next step is to assign the input and/or output signals to the required parameters. This is done through the CLM Configuration Builder and is application-specific. This means that each application will map to the same image point parameters, but the format and conversion process for each specific application will probably vary. Refer to the earlier sections on the CLM Configuration Builder in this manual and the specific application user manual for instructions on mapping your particular device.

Continued on next page

8.2.2.8 Digital Composite Point - 4-State Latched, continued

Command the Point

The Digital point may command an output through the S1REQSTS parameter of the image point. *THE DIGITAL POINT MUST HAVE BEEN MAPPED ALREADY THROUGH THE CLM CONFIGURATION BUILDER.*

The steps to command an output are:

- Select the S1REQSTS parameter

- Select the desired state

- Press ENTER

This will PPS the point and cause the CL to execute, sending the appropriate OPn parameter values to the device.

<p>IMPORTANT! A mapped Digital point will not complete the S1REQSTS store if the associated device (NDT) is Out of Service or in the HARD-FAIL condition. This may be checked at the point by examining the DEV_STA parameter. If the value is OFF, then S1REQSTS stores will not complete and the operator will not see the requested change at the Detail Display.</p>

Clear the Point

After certain alarms or other conditions, it may become necessary to clear an existing alarm(s) and/or line up the S1REQSTS and S1CURSTS parameters. This can be done through the second switch on the Digital point. The steps to perform this action are:

- Select the S2REQSTS parameter

- Select the CLEAR state

- Press ENTER

This action will have the impact of setting the S1REQSTS = S1CURSTS, clearing all existing switch alarms, and re-issuing new alarms based on current conditions.

8.2.2.9 Digital Composite Point, 4-State Pulsed

Operation

The 4 State Digital Composite point emulates digital devices with two inputs and either one or two outputs (e.g. valves). The input states (S1CURSTS) reflect an OPEN, CLOSE, INBTWEEN (\$ord0-\$ord2), and BAD or COMM_ERR (\$ord3) condition. The D1 and D2 parameters reflect the current state of the Digital device, while the OP1, OP2 and OP3 parameters correspond to the pulsed output signals required to drive the Digital device.

NOTE: Although 3 output signals are allowed for the 4-State composite, only 2 states are commandable. This means it takes 3 output signals to map the OPEN state and 3 output signals to map the CLOSE state. The INBTWEEN state cannot be commanded from this point. See the 5-state point definition.

A CL block is attached to this image point that:

- ⇒ Maps the D1 and D2 input signals to one of the ordinal input states (S1CURSTS).
- ⇒ When changed by the operator, maps the ordinal output state (S1REQSTS) to the OP1, OP2 and OP3 output signals, and executes the send to the Digital device.
- ⇒ When an operator change is in progress, monitors the input signals (D1 and D2) for the requested state change against the configured time allowed for the transition (FBTIME), and either alarms or sets the ordinal state (S1CURSTS) to reflect the change.
- ⇒ If configured in ALMUNCHG, alarms when an input state change (D1 and D2) occurs that was not commanded by the operator.
- ⇒ If configured in the alarm options (ALMOPT), issues Off Normal alarms when the input state (D1 and D2) changes off of the normal state (PVNORMAL).

When an operator commands a change (S1REQSTS), the configured output combination is sent to the Digital device. The output feedback signals (FB1, FB2, FB3) are verified against the uncommanded output states (UNC_OP1, UNC_OP2, UNC_OP3). The input signals (D1, D2) are then monitored for the commanded state change. If this state change is not detected before the feedback timer (FBTIME) expires, a CMD_FAIL or CMDDIS alarm is issued.

Alarms may be cleared or the states aligned by selecting the secondary switch (S2REQSTS) option for CLEAR. The CL will set the state back to NOACTION when complete.

Application

Digital device type examples:

- Pump motor monitor and control
- Valve control
- Devices requiring 2 inputs and 1 to 3 latched output signals

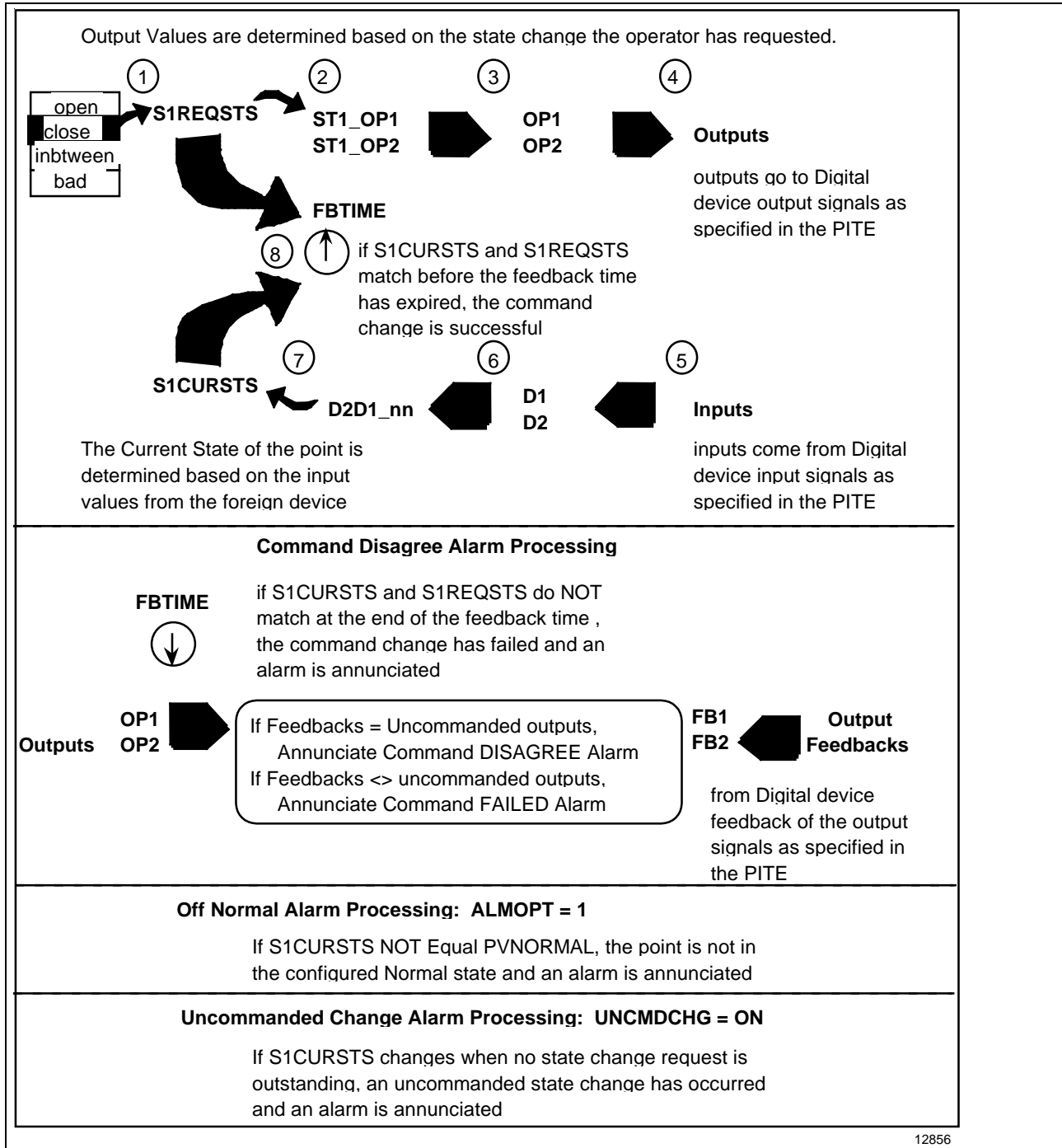
Continued on next page

8.2.2.9 Digital Composite Point - 4-State Pulsed, continued

Operational Flow

Following is a depiction of how a 4-State Digital Composite operates. It shows a Digital Composite with two output signals that are **Pulsed** Outputs with two input signals.

Figure 8-12 Operational Flow for 4-State Pulsed Digital Composite



Continued on next page

8.2.2.9 Digital Composite Point - 4-State Pulsed, continued

I/O Configuration Options

Number of Inputs	Number of Outputs	Max Cmd States
2	1	2
2	2	2
2	3	2

Alarming

The following is a list of alarms that may be issued by this Digital point type. Refer to Table 8-9 for specific alarm definition.

- OUT_SERV
- PT_FAIL
- CFG_ERR_*n*
- INVCMD
- CMDFAIL
- CMDDIS
- NO_MOVE
- UNC_CHG (If configured through ALMUNCHG)
- OFF_NORM (If ALMOPT = 1)
- BAD_PV
- BADSTATE

Build the Point

The 4-State Digital point is based on the AM Switch point. There are two methods of building this point type. Both of these methods use the Data Entity Builder (DEB) and the data entry requirements listed in Table 8-17 below.

- 1) Select the AM SWITCH point type and enter data as listed below.
or
- 2) Copy the DIG4P.EB Exception Build file (included and installed with the Platform release media) to another file for editing.

Edit the new Exception Build file using the data listed below as input for each of the points to be included within the file. Cut and Paste as required.

Perform an Exception Build process on the new file.

Refer to Table 8-10 for complete descriptions of all Digital point parameters.

See Figure 8-13 for an Exception Build file example.

Continued on next page

8.2.2.9 Digital Composite Point - 4-State Pulsed, continued

Table 8-17 4-State Pulsed Digital Point Parameter Entries

Parameter	Entry	Required?	Notes
NAME		Yes	See PKGNAME(4) notes below Application Specific - Refer to the User Manual section for any required packages
UNIT		Yes	
CLSLOTS	1	Yes	
NOPKG	3 or 4	Yes	
NUMSWTCH	2	Yes	
PKGNAME(1)	CLM_DMD3	Yes	
PKGNAME(2)	DIG4CDS	Yes	
PKGNAME(3)	DIG4CDS1	Yes	
PKGNAME(4)	See Notes	-	
S1NSTATE	4	Yes	Enter text for the ordinal 0 (or top) state Enter text for the ordinal 1 state Enter text for the ordinal 2 (between or moving) state Ordinal 3 (or bottom) state text. This state must be the BAD state. Key Access Level for command changes. Default is ENGINEER.
S1STATE(0)	See Notes	Yes	
S1STATE(1)	See Notes	Yes	
S1STATE(2)	See Notes	Yes	
S1STATE(3)	BAD	Yes	
S1ACCLVL	Operator Supervisor Engineer Program	No	
S2NSTATE	2	Yes	Selected state for “resetting” the point. Default state. Selecting has no impact. Key Access Level for clearing switch alarms and setting commanded state = current state. This setting should match the S1ACCLVL. Default is ENGINEER.
S2STATE(0)	CLEAR	Yes	
S2STATE(1)	NOACTION	Yes	
S2ACCLVL	Operator Supervisor Engineer Program	No	
PERIOD	NOPERIOD	Yes	CLM Digital points are PPS'd for execution by data received by exception input or the CLM Point Interface Table (PIT) scheduler.

Continued on next page

8.2.2.9 Digital Composite Point - 4-State Pulsed, continued

Table 8-17 4-State Pulsed Digital Point Parameter Requirements, continued

Parameter	Entry	Required?	Notes
D1_0	\$ord0-\$ord1	Yes	S1CURSTS becomes the ordinal state identified when D1 is OFF.
D1_1	\$ord0-\$ord1	Yes	S1CURSTS becomes the ordinal state identified when D1 is ON.
D2D1_01	\$ord0-\$ord3	Yes	Enter the ordinal state that maps to the condition where bit 1 (D1)=ON and bit 2 (D2)=OFF
D2D1_10	\$ord0-\$ord3	Yes	Enter the ordinal state that maps to the condition where bit 1 (D1)=OFF and bit 2 (D2)=ON
D2D1_11	\$ord0-\$ord3	Yes	Enter the ordinal state that maps to the condition where bit 1 (D1)=ON and bit 2 (D2)=ON
D2D1_00	\$ord0-\$ord3	Yes	Enter the ordinal state that maps to the condition where bit 1 (D1)=OFF and bit 2 (D2)=OFF
NUMCMDST	1-2	Yes	Set to 1 if only ordinal state 0 can be commanded. Set to 2 if both ordinal 0 and 1 can be commanded.
ALMOPT	0-1	No	Default = 0 for operational alarms only. Set to 1 to add Off Normal alarming.
ALMUNCHG	ON or OFF	No	Default = OFF. Set to ON to alarm uncommanded state changes.
PVNORMAL	\$ord0-\$ord1	Yes IF ALMOPT=1	Not required when ALMOPT = 0. Required when ALMOPT = 1.
NODINPTS	2	Yes	Number of Digital Input signals is 2.
FBTIME	See Notes TT.tt	Yes	Feedback Timeouts. Maximum times (in secs) allowed for the commanded state changes, where TT=total time, tt=no movement time. Refer to Table 8-10.
TIMEOP	OFF	Yes	Set to OFF for "Pulsed" outputs.
NODOPTS	2-3	Yes	Number of Digital output signals.
ST0_OP1	ON or OFF	Yes	For ordinal state 0, bit 1. Set to ON for a logical "1", OFF for a logical "0".
ST0_OP2	ON or OFF	Yes IF NODOPTS=2	For ordinal state 0, bit 2. IF NODOPTS = 2, set to ON for a logical "1", OFF for a logical "0".
ST0_OP3	ON or OFF	Yes IF NODOPTS=2	For ordinal state 0, bit 3. IF NODOPTS = 3, set to ON for a logical "1", OFF for a logical "0".

Continued on next page

8.2.2.9 Digital Composite Point - 4-State Pulsed, continued

Table 8-17 4-State Pulsed Digital Point Parameter Requirements, continued

Parameter	Entry	Required?	Notes
ST1_OP1	ON or OFF	Yes	For ordinal state 1, bit 1. Set to ON for a logical "1", OFF for a logical "0".
ST1_OP2	ON or OFF	Yes IF NODOPTS=2	For ordinal state 1, bit 2. IF NODOPTS = 2, set to ON for a logical "1", OFF for a logical "0".
ST1_OP3	ON or OFF	Yes IF NODOPTS=2	For ordinal state 1, bit 3. IF NODOPTS = 3, set to ON for a logical "1", OFF for a logical "0".
UNC_OP1	ON or OFF	Yes	For output bit 1, this is the stable, or non-pulsed, state. Set to On for a logical "1", OFF for a logical "0".
UNC_OP2	ON or OFF	Yes IF NODOPTS=2	For output bit 2, this is the stable, or non-pulsed, state. Set to On for a logical "1", OFF for a logical "0".
UNC_OP3	ON or OFF	Yes IF NODOPTS=3	For output bit 3, this is the stable, or non-pulsed, state. Set to On for a logical "1", OFF for a logical "0".
UNCLRTIM	-1, 0, or 1- <i>n</i>	Yes IF ALMUNCHG= ON	Configurable Clear of the Uncommanded Change alarm. IF ALMUNCHG = ON, set to: -1 for Operator clear only, 0 for automatic clear on next point execution 1- <i>n</i> for automatic clear after <i>n</i> seconds.

Continued on next page

8.2.2.9 Digital Composite Point - 4-State Pulsed, continued

Build Example

The user wants to build a Digital point to emulate a valve. The valve requires a 4 second ON output signal #1 for an OPEN command and a 4 second high output signal #2 for a CLOSE command. The resting, or return state for both output states is OFF. The running state of the valve is monitored through the two limit switches. Input signal #1 (D1) from the CLOSED limit switch and input signal #2 (D2) from the OPENED limit switch. The limit switches are normally open, so the input signals reflect OFF and OFF when the valve is MOVING or inbetween stops. It takes approximately 30 seconds for the valve to completely transition from one stop to the other. If the valve has not started moving after 10 seconds, then there is a problem or a jam. The operator needs to be able to command the valve and it can be started or stopped locally at the valve (uncommanded change). The interface is through a Modbus interface to a PLC and therefore no additional CDS packages are required. Figure 8-13 below shows an example of an Exception Build file (in LCN R500 format) that could be used to create this image point.

Figure 8-13 Build Example for 4-State Pulsed Digital Composite

```
{SCREEN ENTITY VALVE_1()      }
&T SWITCHAM
&N VALVE_1
UNIT = C1
PTDESC ="PUMP 1 CONTROL      "
KEYWORD =" "
CLSLOTS = 1
NOPKG = 3
NUMSWTCH = 2                  {**NOTE: Will cause CL LINK ERROR if not set to 2**}
PRIMMOD = _____
SALMDSC1 =" "
SALMDSC2 =" "
SALMDSC3 =" "
BCLEALPR = LOW                {**NOTE: R500 And Later**}
BCLFALPR = LOW                {**NOTE: R500 And Later**}
CLEALMPR = LOW                {**NOTE: R500 And Later**}
CLFALMPR = LOW                {**NOTE: R500 And Later**}
CNFERRPR = LOW                {**NOTE: R500 And Later**}
SWALM1PR = LOW                {**NOTE: R500 And Later**}
SWALM2PR = LOW                {**NOTE: R500 And Later**}
SWALM3PR = LOW                {**NOTE: R500 And Later**}
ALENBST = ENABLE
PKGNAME(1) ="CLM_DMD3"
PKGNAME(2) ="DIG4CDS "
PKGNAME(3) ="DIG4CDS1"
```

Continued on next page

8.2.2.9 Digital Composite Point - 4-State Pulsed, continued

Figure 8-13 Build Example for 4-State Pulsed Digital Composite, continued

```
S1NSTATE = 4
S1STATES(0) = "OPEN " {**NOTE: User-entered text**}
S1STATES(1) = "CLOSE" {**NOTE: User-entered text**}
S1STATES(2) = "MOVING "
S1STATES(3) = "BAD "
S1ACCLVL = OPERATOR
S1CURSTS = CLOSE
S1REQSTS = CLOSE
S2NSTATE = 2
S2STATES(0) = "CLEAR "
S2STATES(1) = "NOACTION"
S2ACCLVL = OPERATOR
S2CURSTS = NOACTION
S2REQSTS = NOACTION
$IPPASN = OFF
PERIOD = NOPERIOD {**NOTE: Scheduled and driven by associated PIT**}
CLM_PORT = " "
CLM_DEV = " "
CLM_PIT = " "
NODINPTS = 2
D1_0 = $ORD0 {**NOTE: Valve is OPEN when state is logical zero**}
D1_1 = $ORD1
D2D1_01 = $ORD1 {**CLOSE**}
D2D1_10 = $ORD0 {**OPEN**}
D2D1_00 = $ORD2 {**MOVING**}
D2D1_11 = $ORD3 {**BAD**}
NUMCMDST = 2
ALMOPT = 0
ALMUNCHG = ON
PVNORMAL = $ORD1
FBTIME = 30.10
TIMEOP = ON {**NOTE: MUST be ON for Pulsed output signals**}
NODOPTS = 2
ST0_OP1 = OFF {**NOTE: OPEN state sets output signal #2 to ON**}
ST0_OP2 = ON
ST0_OP3 = OFF
ST1_OP1 = ON {**NOTE: CLOSE state sets output signal #1 to ON**}
ST1_OP2 = OFF
ST1_OP3 = OFF
UNC_OP1 = OFF
UNC_OP2 = OFF
UNC_OP3 = OFF
UNCLRTIM = -1
UNCCLEAR = OFF
DEV_STA = OFF
PT_FAIL = OFF
CMDCHG = OFF
```

8.2.2.9 Digital Composite Point - 4-State Pulsed, continued

Figure 8-13 Build Example for 4-State Pulsed Digital Composite, continued

```
ALMTYPE1 = A_NONE  
ALMTYPE2 = A_NONE  
ALMTYPE3 = A_NONE  
INITLIZD = OFF  
TIMER   = 0  
DATA(1) = 0  
DATA(2) = 0  
DATA(3) = 0  
DATA(4) = 0  
DATA(5) = 0  
DATA(6) = 0  
DATA(7) = 0  
DATA(8) = 0  
DATA(9) = 0  
DATA(10) = 0
```

Continued on next page

8.2.2.9 Digital Composite Point - 4-State Pulsed, continued

Attach the CL Block

All CLM Digital points require a CL Block. The 4-State Digital Composite uses the DIG4CL.AO block included with the CLM Platform release media. Use the EC file included to link the block to the 4-State point.

From the COMMAND PROCESSOR window of the Engineering Personality:

```
EC DIG4.EC pppppppp
where
pppppppp is the 4-State point name
```

NOTE: 1) Be sure that your pathnames are set for the CL SOURCE/OBJ to point to the proper directory.
2) When executing this EC file for the first time, it is normal to get an error on the UNLINK since the block has not yet been attached.

Link Caution!

If the number of switches (NUMSWTCH) on the image point is not set to 2, the following error will be issued by the Linker.

Error during attempt to get parameter information; PA Status: END OF USER SPEC

Correct and rebuild the image point before attempting to re-link.

Activate the Point

At the Detail Display of the Digital point, set the PTEXECST = ACTIVE. You may need to PROCESS the point again to clear the RESTART state to None. The S1REQSTS and S1CURSTS should agree at this point, but if not, select the S2REQSTS, select the CLEAR option and press ENTER.

Map to the Device (See Application User Manual)

The build of the image is complete from the AM side. The next step is to assign the input and/or output signals to the required parameters. This is done through the CLM Configuration Builder and is application-specific. This means that each application will map to the same image point parameters, but the format and conversion process for each specific application will probably vary. Refer to the earlier sections on the CLM Configuration Builder in this manual and the specific application user manual for instructions on mapping your particular device.

Continued on next page

8.2.2.9 Digital Composite Point - 4-State Pulsed, continued

Command the Point

The Digital point may command an output through the S1REQSTS parameter of the image point. *THE DIGITAL POINT MUST HAVE BEEN MAPPED ALREADY THROUGH THE CLM CONFIGURATION BUILDER.*

The steps to command an output are:

- Select the S1REQSTS parameter

- Select the desired state

- Press ENTER

This will PPS the point and cause the CL to execute, sending the appropriate OPn parameter values to the device.

<p>IMPORTANT! A mapped Digital point will not complete the S1REQSTS store if the associated device (NDT) is Out of Service or in the HARD-FAIL condition. This may be checked at the point by examining the DEV_STA parameter. If the value is OFF, then S1REQSTS stores will not complete and the operator will not see the requested change at the Detail Display.</p>

Clear the Point

After certain alarms or other conditions, it may become necessary to clear an existing alarm(s) and/or line up the S1REQSTS and S1CURSTS parameters. This can be done through the second switch on the Digital point. The steps to perform this action are:

- Select the S2REQSTS parameter

- Select the CLEAR state

- Press ENTER

This action will have the impact of setting the S1REQSTS = S1CURSTS, clearing all existing switch alarms, and re-issuing new alarms based on current conditions.

Operation

The 4 State Digital Input point monitors digital devices with a two input signals. The input state (S1CURSTS) is a mapped 2-bit configuration reflecting the descriptive value of the D1 and D2 parameters. The last ordinal state reflects the BAD or COMM_ERR (\$ord3) condition.

A CL block is attached to this image point that:

- ⇒ Maps the D1 and D2 input signals to one of the ordinal input states (S1CURSTS).
- ⇒ If configured in the alarm options (ALMOPT), issues Off Normal alarms when the input state (D1 and D2) changes off of the normal state (PVNORMAL).
- ⇒ If configured in the alarm options (ALMOPT), issues Change-of-State alarms when the input state (D1 and D2) changes.

Although this point type is not commandable, the S1REQSTS may be configured to track the current state (S1CURSTS). This is accomplished by use of the second switch. By selecting the S2REQSTS, the switch #1 requested state may be configured to TRACK or NO_TRACK the current state.

Application

Digital device type examples:

- Valve monitoring
- Fan motor monitoring
- Digital signals from devices requiring 2 inputs

Continued on next page

8.2.2.10 Digital Input Point, 4-State, continued

Alarming

The following is a list of alarms that may be issued by this Digital point type. Refer to Table 8-9 for specific alarm definition.

- OUT_SERV
- PT_FAIL
- CFG_ERR_#
- OFF_NORM (If ALMOPT = 1)
- CHGSTATE (If ALMOPT = 2)

Build the Point

The 4-State Digital input point is based on the AM Switch point. There are two methods of building this point type. Both of these methods use the Data Entity Builder (DEB) and the data entry requirements listed in Table 8-18 below.

- 1) Select the AM SWITCH point type and enter data as listed below.

or

- 2) Copy the DIN4.EB Exception Build file (included and installed with the Platform release media) to another file for editing.

Edit the new Exception Build file using the data listed below as input for each of the points to be included within the file. Cut and Paste as required.

Perform an Exception Build process on the new file.

Refer to Table 8-10 for complete descriptions of all Digital point parameters.

See Figure 8-14 for an Exception Build file example.

Continued on next page

8.2.2.10 Digital Input Point - 4-State, continued

Table 8-18 4-State Digital Input Parameter Entries

Parameter	Entry	Required?	Notes
NAME		Yes	See PKGNAME(3) notes below
UNIT		Yes	
CLSLOTS	1	Yes	
NOPKG	2 or 3	Yes	
NUMSWTCH	2	Yes	
PKGNAME(1)	CLM_DMD3	Yes	
PKGNAME(2)	DIN4CDS	Yes	
PKGNAME(3)	See Notes	-	
S1NSTATE	3	Yes	Enter text for the ordinal 0 (or top) state
S1STATE(0)	See Notes	Yes	
S1STATE(1)	See Notes	Yes	
S1STATE(2)	See Notes	Yes	
S1STATE(3)	BAD	Yes	
S1ACCLVL	Operator Supervisor Engineer Program	No	
S2NSTATE	2	Yes	Selected state for "tracking" option
S2STATE(0)	TRACK	Yes	
S2STATE(1)	NO_TRACK	Yes	
S2ACCLVL	Operator Supervisor Engineer Program	No	
PERIOD	NOPERIOD	Yes	CLM Digital points are PPS'd for execution by data received by exception input or the CLM Point Interface Table (PIT) scheduler.

Continued on next page

8.2.2.10 Digital Input Point - 4-State, continued

Table 8-18 4-State Digital Input Parameter Entries, continued

Parameter	Entry	Required?	Notes
D2D1_01	\$ord0-\$ord3	Yes	Enter the ordinal state that maps to the condition where bit 1 (D1)=ON and bit 2 (D2)=OFF
D2D1_10	\$ord0-\$ord3	Yes	Enter the ordinal state that maps to the condition where bit 1 (D1)=OFF and bit 2 (D2)=ON
D2D1_11	\$ord0-\$ord3	Yes	Enter the ordinal state that maps to the condition where bit 1 (D1)=ON and bit 2 (D2)=ON
D2D1_00	\$ord0-\$ord3	Yes	Enter the ordinal state that maps to the condition where bit 1 (D1)=OFF and bit 2 (D2)=OFF
ALMOPT	0-2,11-12	No	Alarming Options: 0 = Configuration Alarms Only 1 = Off-Normal Alarming (all states) 2 = Chg-State Alarming (all states) 11 = Off-Normal Alarming (Limits only) 12 = Chg-State Alarming (Limits only)
PVNORMAL	\$ord0-\$ord1	Yes IF ALMOPT=1,11	Not required when ALMOPT = 0, 2, 12. Required when ALMOPT = 1 or 11.

Continued on next page

8.2.2.10 Digital Input Point - 4-State, continued

Build Example

The user wants to build a Digital point to monitor a valve. The running state of the valve is monitored through the two limit switches. Input signal #1 (D1) from the CLOSED limit switch and input signal #2 (D2) from the OPENED limit switch. The limit switches are normally open, so the input signals reflect OFF and OFF when the valve is MOVING or inbetween stops. The normal state of the valve is CLOSED. The interface is through a Modbus interface to a PLC and therefore no additional CDS packages are required. Figure 8-14 below shows an example of an Exception Build file (in LCN R500 format) that could be used to create this image point.

Figure 8-14 Build Example for 4-State Digital Input

```
{-----FILE: DI3.EB-----}
{SCREEN ENTITY VALVE_1() }
&T SWITCHAM
&N VALVE_1
UNIT = C1
PTDESC ="VALVE 1 OPERATION "
KEYWORD =" "
CLSLOTS = 1
NOPKG = 2
NUMSWTCH = 2          {**NOTE: Will cause CL LINK ERROR if not set to 2**}
PRIMMOD = _____
SALMDSC1 =" "
SALMDSC2 =" "
SALMDSC3 =" "
BCLEALPR = LOW        {**NOTE: R500 And Later**}
BCLFALPR = LOW        {**NOTE: R500 And Later**}
CLEALMPR = LOW        {**NOTE: R500 And Later**}
CLFALMPR = LOW        {**NOTE: R500 And Later**}
CNFERRPR = LOW        {**NOTE: R500 And Later**}
SWALM1PR = LOW        {**NOTE: R500 And Later**}
SWALM2PR = LOW        {**NOTE: R500 And Later**}
SWALM3PR = LOW        {**NOTE: R500 And Later**}
ALENBST = ENABLE
PKGNAME(1) ="CLM_DMD3"
PKGNAME(2) ="DIN4CDS "
S1NSTATE = 4
S1STATES(0) ="OPENED"  {**NOTE: User-entered text**}
S1STATES(1) ="CLOSED"  {**NOTE: User-entered text**}
S1STATES(2) ="MOVING"
S1STATES(3) = "BAD"
S1ACCLVL =ENGINEER
S1CURSTS = CLOSED
S1REQSTS = CLOSED
```

Continued on next page

8.2.2.10 Digital Input Point - 4-State, continued

Figure 8-14 Build Example for 4-State Digital Input, continued

```
S2NSTATE = 2
S2STATES(0) = "TRACK "
S2STATES(1) = "NO_TRACK"
S2ACCLVL = OPERATOR
S2CURSTS = NO_TRACK
S2REQSTS = NO_TRACK
$IPPASN = OFF
PERIOD = NOPERIOD      {**NOTE: Scheduled and driven by associated PIT**}
CLM_PORT = " "
CLM_DEV = " "
CLM_PIT = " "
D2D1_00 = $ORD2
D2D1_01 = $ORD1
D2D1_10 = $ORD0
D2D1_11 = $ORD3
ALMOPT = 0             {**NOTE: Off-Normal alarming option is not selected here**}
PVNORMAL = $ORD1       {**NOTE: CLOSED is the normal state**}
DEV_STA = OFF
PT_FAIL = OFF
```

Continued on next page

8.2.2.10 Digital Input Point - 4-State, continued

Attach the CL Block

All CLM Digital points require a CL Block. The 4-State Digital Input uses the DIN4CL.AO block included with the CLM Platform release media. Use the EC file included to link the block to the 4-State point.

From the COMMAND PROCESSOR window of the Engineering Personality:

EC DIN4.EC *pppppppp*

where

pppppppp is the 4-State point name

NOTE: 1) Be sure that your pathnames are set for the CL SOURCE/OBJ to point to the proper directory.
2) When executing this EC file for the first time, it is normal to get an error on the UNLINK since the block has not yet been attached.

Link Caution!

If the number of switches (NUMSWTCH) on the image point is not set to 2, the following error will be issued by the Linker.

Error during attempt to get parameter information; PA Status: END OF USER SPEC

Correct and rebuild the image point before attempting to re-link.

Activate the Point

At the Detail Display of the Digital point, set the PTEXECST = ACTIVE. You may need to PROCESS the point again to clear the RESTART state to None.

Map to the Device (See Application User Manual)

The build of the image is complete from the AM side. The next step is to assign the input and/or output signals to the required parameters. This is done through the CLM Configuration Builder and is application-specific. This means that each application will map to the same image point parameters, but the format and conversion process for each specific application will probably vary. Refer to the earlier sections on the CLM Configuration Builder in this manual and the specific application user manual for instructions on mapping your particular device.

Continued on next page

8.2.2.10 Digital Input Point - 4-State, continued

Tracking Option

Since there is no commanded output on the Digital Input point, the command state (S1REQSTS) may be configured to follow, or track, the current input state (S1CURSTS). This option is configured through the second switch (S2REQSTS) on the Digital Input point. Configuration is as follows:

- | | |
|----------|--|
| TRACK | S1REQSTS set equal to S1CURSTS on point update. |
| NO_TRACK | S1REQSTS is not changed on point update. Note that the S1REQSTS may be used to display the normal state in this case if the off-normal alarming is configured (ALMOPT =1 or 11). |

Off-Normal and Change-of-State alarming are not affected by the tracking option.

Operation

The 4 State Digital Output point emulates digital devices with no inputs and either two or three outputs (e.g. valves). The output states (S1REQSTS/S1CURSTS) reflect an ON, OFF, MOVING (\$ord0-\$ord2), and BAD or COMM_ERR (\$ord3) condition. The OP1, OP2 and OP3 parameters correspond to the latched output signals required to drive the Digital device, while the FB1, FB2 and FB3 parameters monitor the output feedback signals from the device.

A CL block is attached to this image point that:

- ⇒ When changed by the operator, maps the ordinal output state (S1REQSTS) to the OP1, OP2 and OP3 output signals, and executes the send to the Digital device.
- ⇒ When an operator change is in progress, monitors the feedback signals (FB1, FB2, FB3) for the requested state change against the configured time allowed for the transition (FBTIME), and either alarms or sets the ordinal state (S1CURSTS) to reflect the change.

NOTE: Although all three output/feedback signals are available on the 4-State point, only two combinations of the three bits are allowed. This means that there are only two (2) commandable states.

When an operator commands a change (S1REQSTS), the configured output combination is sent to the Digital device. The output feedback signals (FB1, FB2, FB3) are checked to verify that the output signals (OP1, OP2, OP3) were properly received and processed by the Digital device. If this state change is not detected before the feedback timer (FBTIME) expires, a CMD_FAIL or CMDDIS alarm is issued.

Alarms may be cleared or the states aligned by selecting the secondary switch (S2REQSTS) option for CLEAR. The CL will set the state back to NOACTION when complete.

Application

Digital device type examples:

- Motor-Operated-Valves
- Devices requiring 2 or 3 latched output signals with feedbacks only

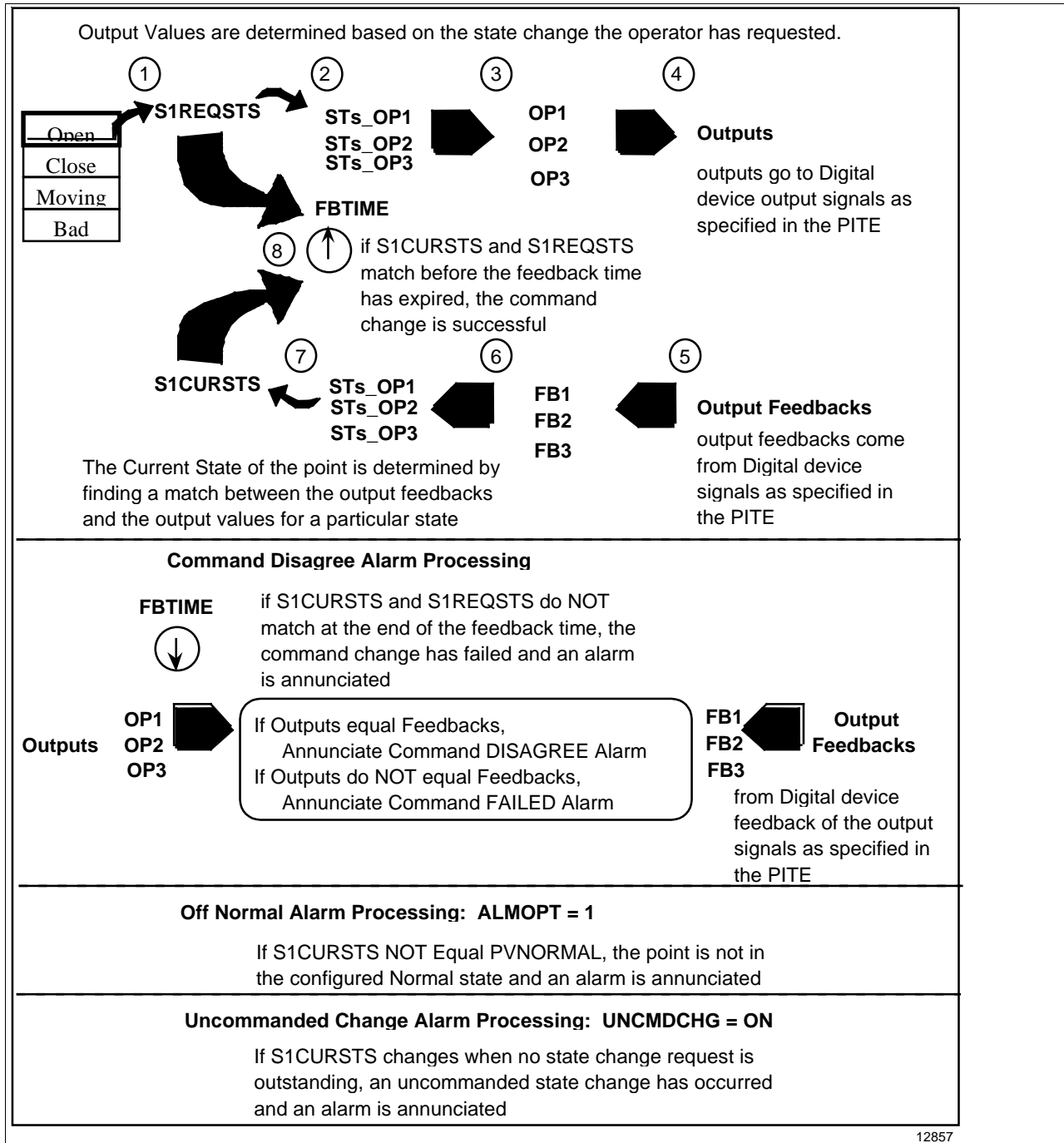
Continued on next page

8.2.2.11 Digital Output Point - 4-State Latched, continued

Operational Flow

Following is a depiction of how a 4-State digital output operates. It shows two output signals that are **Latched** Outputs with no input signals.

Figure 8-15 Operational Flow for 4-State Latched Digital Output



Continued on next page

8.2.2.11 Digital Output Point - 4-State Latched, continued

Alarming

The following is a list of alarms that may be issued by this Digital point type. Refer to Table 8-9 for specific alarm definition.

- OUT_SERV
 - PT_FAIL
 - CFG_ERR_*n*
 - INVCMD
 - CMDFAIL
 - CMDDIS
 - UNC_CHG (If configured through ALMUNCHG)
 - BAD_PV
 - BADSTATE
-

Build the Point

The 4-State Digital point is based on the AM Switch point. There are two methods of building this point type. Both of these methods use the Data Entity Builder (DEB) and the data entry requirements listed in Table 8-19 below.

- 1) Select the AM SWITCH point type and enter data as listed below.
or
- 2) Copy the DOUT4L.EB Exception Build file (included and installed with the Platform release media) to another file for editing.

Edit the new Exception Build file using the data listed below as input for each of the points to be included within the file. Cut and Paste as required.

Perform an Exception Build process on the new file.

Refer to Table 8-10 for complete descriptions of all Digital point parameters.

See Figure 8-16 for an Exception Build file example.

Continued on next page

8.2.2.11 Digital Output Point - 4-State Latched, continued

Table 8-19 4-State Latched Digital Output Parameter Entries

Parameter	Entry	Required?	Notes
NAME		Yes	See PKGNAME(4) notes below Application Specific - Refer to the User Manual section for any required packages
UNIT		Yes	
CLSLOTS	1	Yes	
NOPKG	3 or 4	Yes	
NUMSWTCH	2	Yes	
PKGNAME(1)	CLM_DMD3	Yes	
PKGNAME(2)	DIG4CDS	Yes	
PKGNAME(3)	DIG4CDS1	Yes	
PKGNAME(4)	See Notes	-	
S1NSTATE	4	Yes	Enter text for the ordinal 0 (or top) state Enter text for the ordinal 1 state Enter text for the ordinal 2 (between or moving) state Ordinal 3 (or bottom) state text. This state must be the BAD state. Key Access Level for command changes. Default is ENGINEER.
S1STATE(0)	See Notes	Yes	
S1STATE(1)	See Notes	Yes	
S1STATE(2)	See Notes	Yes	
S1STATE(3)	BAD	Yes	
S1ACCLVL	Operator Supervisor Engineer Program	No	
S2NSTATE	2	Yes	Selected state for “resetting” the point. Default state. Selecting has no impact. Key Access Level for clearing switch alarms and setting commanded state = current state. This setting should match the S1ACCLVL. Default is ENGINEER.
S2STATE(0)	CLEAR	Yes	
S2STATE(1)	NOACTION	Yes	
S2ACCLVL	Operator Supervisor Engineer Program	No	
PERIOD	NOPERIOD	Yes	CLM Digital points are PPS'd for execution by data received by exception input or the CLM Point Interface Table (PIT) scheduler.

Continued on next page

8.2.2.11 Digital Output Point - 4-State Latched, continued

Table 8-19 4-State Latched Digital Output Parameter Entries, continued

Parameter	Entry	Required?	Notes
D1_0		No	Not used for Digital Outputs
D1_1		No	Not used for Digital Outputs
D2D1_01		No	Not used for Digital Outputs
D2D1_10		No	Not used for Digital Outputs
D2D1_11		No	Not used for Digital Outputs
D2D1_00		No	Not used for Digital Outputs
NUMCMDST	1-2	Yes	Set to 1 if only ordinal state 0 can be commanded. Set to 2 if both ordinal 0 and 1 can be commanded.
ALMOPT	0-1	No	Default = 0 for operational alarms only. Set to 1 to add Off Normal alarming.
ALMUNCHG	ON or OFF	No	Default = OFF. Set to ON to alarm uncommanded state changes.
PVNORMAL		No	Not used for Digital Outputs
NODINPTS	0	Yes	Must be zero (0) for Digital Outputs
FBTIME	See Notes TT,tt	Yes	Feedback Timeouts. Maximum times (in secs) allowed for the commanded state changes, where TT=total time, tt=no movement time. Refer to Table 8-10.
TIMEOP	OFF	Yes	Set to OFF for "Latched" outputs.
NODOPTS	2-3	Yes	Number of Digital output signals.
ST0_OP1	ON or OFF	Yes	For ordinal state 0, bit 1. Set to ON for a logical "1", OFF for a logical "0".
ST0_OP2	ON or OFF	Yes IF NODOPTS=2	For ordinal state 0, bit 2. IF NODOPTS = 2, set to ON for a logical "1", OFF for a logical "0".
ST0_OP3	ON or OFF	Yes IF NODOPTS=3	For ordinal state 0, bit 3. IF NODOPTS = 3, set to ON for a logical "1", OFF for a logical "0".

Continued on next page

8.2.2.11 Digital Output Point - 4-State Latched, continued

Table 8-19 4-State Latched Digital Output Parameter Entries, continued

Parameter	Entry	Required?	Notes
ST1_OP1	ON or OFF	Yes	For ordinal state 1, bit 1. Set to ON for a logical "1", OFF for a logical "0".
ST1_OP2	ON or OFF	Yes IF NODOPTS=2	For ordinal state 1, bit 2. IF NODOPTS = 2, set to ON for a logical "1", OFF for a logical "0".
ST1_OP3	ON or OFF	Yes IF NODOPTS=2	For ordinal state 1, bit 3. IF NODOPTS = 3, set to ON for a logical "1", OFF for a logical "0".
UNCLRTIM	-1, 0, or 1- <i>n</i>	Yes IF ALMUNCHG= ON	Configurable Clear of the Uncommanded Change alarm. IF ALMUNCHG = ON, set to: -1 for Operator clear only, 0 for automatic clear on next point execution 1- <i>n</i> for automatic clear after <i>n</i> seconds.

Continued on next page

8.2.2.11 Digital Output Point - 4-State Latched, continued

Build Example

A Digital point needs to be built to open and close valve #1. The valve requires output signal #1 = ON for OPEN and output signal #2 = ON for CLOSE. There are no input signals to monitor the running state, so the output feedbacks must be checked to be sure that the command was accepted by the Digital device. The response time from the Digital device to set the commanded state is approximately 10 seconds. The operator needs to be able to command the valve and wants to know if the valve is opened or closed at the valve locally (uncommanded change). The interface is Modbus and therefore requires no additional CDS packages. Figure 8-16 below shows an example of an Exception Build file (in LCN R500 format) that could be used to create this image point.

Figure 8-16 Build Example for 4-State Latched Digital Output

```
{SCREEN ENTITY MOV_1() }
&T SWITCHAM
&N MOV_1
UNIT = C1
PTDESC ="VALVE 1 CONTROL  "
KEYWORD ="  "
CLSLOTS = 1
NOPKG = 3
NUMSWTCH = 2          {**NOTE: Will cause CL LINK ERROR if not set to 2**}
PRIMMOD = _____
SALMDSC1 ="  "
SALMDSC2 ="  "
SALMDSC3 ="  "
BCLEALPR = LOW        {**NOTE: R500 And Later**}
BCLFALPR = LOW        {**NOTE: R500 And Later**}
CLEALMPR = LOW        {**NOTE: R500 And Later**}
CLFALMPR = LOW        {**NOTE: R500 And Later**}
CNFERRPR = LOW        {**NOTE: R500 And Later**}
SWALM1PR = LOW        {**NOTE: R500 And Later**}
SWALM2PR = LOW        {**NOTE: R500 And Later**}
SWALM3PR = LOW        {**NOTE: R500 And Later**}
ALENBST = ENABLE
PKGNAME(1) ="CLM_DMD3"
PKGNAME(2) ="DIG4CDS "
PKGNAME(3) ="DIG4CDS1"
S1NSTATE = 4
S1STATES(0) ="OPEN  "  {**NOTE: User-entered text**}
S1STATES(1) ="CLOSE  " {**NOTE: User-entered text**}
S1STATES(2) ="MOVING "
S1STATES(3) ="BAD   "
SIACCLVL = OPERATOR
```

Continued on next page

8.2.2.11 Digital Output Point - 4-State Latched, continued

Figure 8-16 Build Example for 4-State Latched Digital Output , continued

```
S1CURSTS = OPEN
S1REQSTS = OPEN
S2NSTATE = 2
S2STATES(0) = "CLEAR "
S2STATES(1) = "NOACTION"
S2ACCLVL = OPERATOR
S2CURSTS = NOACTION
S2REQSTS = NOACTION
$IPPASN = OFF
PERIOD = NOPERIOD          {**NOTE: Scheduled and driven by associated PIT**}
CLM_PORT = " "
CLM_DEV = " "
CLM_PIT = " "
NODINPTS = 0
NUMCMDST = 2
ALMOPT = 0
ALMUNCHG = ON
FBTIME = 20.10
TIMEOP = OFF
NODOPTS = 2
ST0_OP1 = ON               {**NOTE: OPEN state sets output signal #1 to ON**}
ST0_OP2 = OFF
ST1_OP1 = OFF
ST1_OP2 = ON               {**NOTE: CLOSE state sets output signal #2 to ON**}
UNC_OP1 = OFF
UNC_OP2 = OFF
UNCLRTIM = -1
UNCCLEAR = OFF
DEV_STA = OFF
PT_FAIL = OFF
CMDCHG = OFF
ALMTYPE1 = A_NONE
ALMTYPE2 = A_NONE
ALMTYPE3 = A_NONE
INITLIZD = OFF
TIMER = 0
DATA(1) = 0
DATA(2) = 0
DATA(3) = 0
DATA(4) = 0
DATA(5) = 0
DATA(6) = 0
DATA(7) = 0
DATA(8) = 0
DATA(9) = 0
DATA(10) = 0
```

8.2.2.11 Digital Output Point - 4-State Latched, continued

Attach the CL Block

All CLM Digital points require a CL Block. The 4-State Digital Output uses the DIG4CL.AO block included with the CLM Platform release media. Use the EC file included to link the block to the 4-State point.

From the COMMAND PROCESSOR window of the Engineering Personality:

```
EC DIG4.EC pppppppp
      where
      pppppppp is the 4-State point name
```

NOTE: 1) Be sure that your pathnames are set for the CL SOURCE/OBJ to point to the proper directory.
2) When executing this EC file for the first time, it is normal to get an error on the UNLINK since the block has not yet been attached.

Link Caution!

If the number of switches (NUMSWTCH) on the image point is not set to 2, the following error will be issued by the Linker.

Error during attempt to get parameter information; PA Status: END OF USER SPEC

Correct and rebuild the image point before attempting to re-link.

Activate the Point

At the Detail Display of the Digital point, set the PTEXECST = ACTIVE. You may need to PROCESS the point again to clear the RESTART state to NONE. The S1REQSTS and S1CURSTS should agree at this point, but if not, select the S2REQSTS, select the CLEAR option and press ENTER.

Map to the Device (See Application User Manual)

The build of the image is complete from the AM side. The next step is to assign the input and/or output signals to the required parameters. This is done through the CLM Configuration Builder and is application-specific. This means that each application will map to the same image point parameters, but the format and conversion process for each specific application will probably vary. Refer to the earlier sections on the CLM Configuration Builder in this manual and the specific application user manual for instructions on mapping your particular device.

Continued on next page

8.2.2.11 Digital Output Point - 4-State Latched, continued

Command the Point

The Digital point may command an output through the S1REQSTS parameter of the image point. *THE DIGITAL POINT MUST HAVE BEEN MAPPED ALREADY THROUGH THE CLM CONFIGURATION BUILDER.*

The steps to command an output are:

- Select the S1REQSTS parameter

- Select the desired state

- Press ENTER

This will PPS the point and cause the CL to execute, sending the appropriate OPn parameter values to the device.

<p>IMPORTANT! A mapped Digital point will not complete the S1REQSTS store if the associated device (NDT) is Out of Service or in the HARD-FAIL condition. This may be checked at the point by examining the DEV_STA parameter. If the value is OFF, then S1REQSTS stores will not complete and the operator will not see the requested change at the Detail Display.</p>

Clear the Point

After certain alarms or other conditions, it may become necessary to clear an existing alarm(s) and/or line up the S1REQSTS and S1CURSTS parameters. This can be done through the second switch on the Digital point. The steps to perform this action are:

- Select the S2REQSTS parameter

- Select the CLEAR state

- Press ENTER

This action will have the impact of setting the S1REQSTS = S1CURSTS, clearing all existing switch alarms, and re-issuing new alarms based on current conditions.

Operation

The 4 State Digital Output point emulates digital devices with no inputs and either two or three outputs (e.g. valves). The output states (S1REQSTS/S1CURSTS) reflect an ON, OFF, MOVING (\$ord0-\$ord2), and BAD or COMM_ERR (\$ord3) condition. The OP1, OP2 and OP3 parameters correspond to the Pulsed output signals required to drive the Digital device, while the FB1, FB2 and FB3 parameters monitor the output feedback signals from the device.

A CL block is attached to this image point that:

- ⇒ When changed by the operator, maps the ordinal output state (S1REQSTS) to the OP1, OP2 and OP3 output signals, and executes the send to the Digital device.
- ⇒ When an operator change is in progress, monitors the feedback signals (FB1, FB2, FB3) for the requested state change against the configured time allowed for the transition (FBTIME), and either alarms or sets the ordinal state (S1CURSTS) to reflect the change.

NOTE: Although all three output/feedback signals are available on the 4-State point, only two combinations of the three bits are allowed. This means that there are only two (2) commandable states.

When an operator commands a change (S1REQSTS), the configured output combination is sent to the Digital device. The output feedback signals (FB1, FB2, FB3) are compared against the uncommanded state values (UNC_OP1, UNC_OP2, UNC_OP3) to verify that the driving device has returned back to the normal or “stable” states. If this state change is not detected before the feedback timer (FBTIME) expires, a CMD_FAIL or CMDDIS alarm is issued.

Alarms may be cleared or the states aligned by selecting the secondary switch (S2REQSTS) option for CLEAR. The CL will set the state back to NOACTION when complete.

Application

Digital device type examples:

- Motor-Operated-Valves
- Devices requiring 2 or 3 Pulsed output signals with feedbacks only

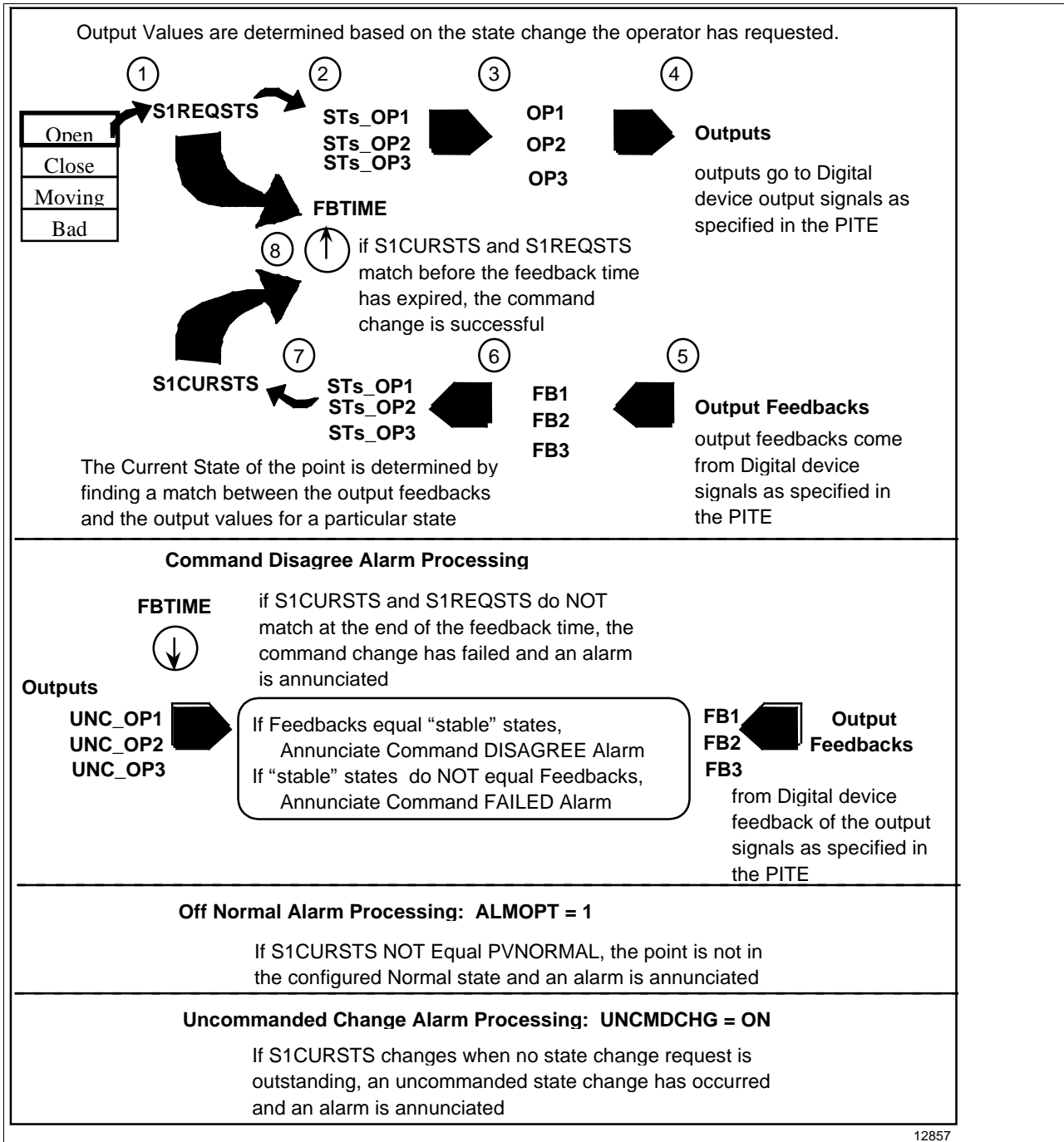
Continued on next page

8.2.2.12 Digital Output Point - 4-State Pulsed, continued

Operational Flow

Following is a depiction of how a 4-State Digital Output operates. It shows two output signals that are **Pulsed** Outputs with no input signals.

Figure 8-17 Operational Flow for 4-State Pulsed Digital Output



Continued on next page

8.2.2.12 Digital Output Point - 4-State Pulsed, continued

Alarming

The following is a list of alarms that may be issued by this Digital point type. Refer to Table 8-9 for specific alarm definition.

- OUT_SERV
 - PT_FAIL
 - CFG_ERR_*n*
 - INVCMD
 - CMDFAIL
 - CMDDIS
 - UNC_CHG (If configured through ALMUNCHG)
 - BAD_PV
 - BADSTATE
-

Build the Point

The 4-state Digital point is based on the AM Switch point. There are two methods of building this point type. Both of these methods use the Data Entity Builder (DEB) and the data entry requirements listed in Table 8-20 below.

- 1) Select the AM SWITCH point type and enter data as listed below.
or
- 2) Copy the DOUT4P.EB Exception Build file (included and installed with the Platform release media) to another file for editing.

Edit the new Exception Build file using the data listed below as input for each of the points to be included within the file. Cut and Paste as required.

Perform an Exception Build process on the new file.

Refer to Table 8-10 for complete descriptions of all Digital point parameters.

See Figure 8-18 for an Exception Build file example.

Continued on next page

8.2.2.12 Digital Output Point - 4-State Pulsed, continued

Table 8-20 4-State Pulsed Digital Output Parameter Entries

Parameter	Entry	Required?	Notes
NAME		Yes	See PKGNAME(4) notes below Application Specific - Refer to the User Manual section for any required packages
UNIT		Yes	
CLSLOTS	1	Yes	
NOPKG	3 or 4	Yes	
NUMSWTCH	2	Yes	
PKGNAME(1)	CLM_DMD3	Yes	
PKGNAME(2)	DIG4CDS	Yes	
PKGNAME(3)	DIG4CDS1	Yes	
PKGNAME(4)	See Notes	-	
S1NSTATE	4	Yes	Enter text for the ordinal 0 (or top) state Enter text for the ordinal 1 state Enter text for the ordinal 2 (between or moving) state Ordinal 3 (or bottom) state text. This state must be the BAD state. Key Access Level for command changes. Default is ENGINEER.
S1STATE(0)	See Notes	Yes	
S1STATE(1)	See Notes	Yes	
S1STATE(2)	See Notes	Yes	
S1STATE(3)	BAD	Yes	
S1ACCLVL	Operator Supervisor Engineer Program	No	
S2NSTATE	2	Yes	Selected state for “resetting” the point. Default state. Selecting has no impact. Key Access Level for clearing switch alarms and setting commanded state = current state. This setting should match the S1ACCLVL. Default is ENGINEER.
S2STATE(0)	CLEAR	Yes	
S2STATE(1)	NOACTION	Yes	
S2ACCLVL	Operator Supervisor Engineer Program	No	
PERIOD	NOPERIOD	Yes	CLM Digital points are PPS'd for execution by data received by exception input or the CLM Point Interface Table (PIT) scheduler.

Continued on next page

8.2.2.12 Digital Output Point - 4-State Pulsed, continued

Table 8-20 4-State Pulsed Digital Output Parameter Entries, continued

Parameter	Entry	Required?	Notes
D1_0		No	Not used for Digital Outputs
D1_1		No	Not used for Digital Outputs
D2D1_01		No	Not used for Digital Outputs
D2D1_10		No	Not used for Digital Outputs
D2D1_11		No	Not used for Digital Outputs
D2D1_00		No	Not used for Digital Outputs
NUMCMDST	1-2	Yes	Set to 1 if only ordinal state 0 can be commanded. Set to 2 if both ordinal 0 and 1 can be commanded.
ALMOPT	0-1	No	Default = 0 for operational alarms only. Set to 1 to add Off Normal alarming.
ALMUNCHG	ON or OFF	No	Default = OFF. Set to ON to alarm uncommanded state changes.
PVNORMAL		No	Not used for Digital Outputs
NODINPTS	0	Yes	Must be zero (0) for Digital Outputs
FBTIME	See Notes TT,tt	Yes	Feedback Timeouts. Maximum times (in secs) allowed for the commanded state changes, where TT=total time, tt=no movement time. Refer to Table 8-10.
TIMEOP	ON	Yes	Set to ON for "Pulsed" outputs.
NODOPTS	2-3	Yes	Number of Digital output signals.
ST0_OP1	ON or OFF	Yes	For ordinal state 0, bit 1. Set to ON for a logical "1", OFF for a logical "0".
ST0_OP2	ON or OFF	Yes IF NODOPTS=2	For ordinal state 0, bit 2. IF NODOPTS = 2, set to ON for a logical "1", OFF for a logical "0".
ST0_OP3	ON or OFF	Yes IF NODOPTS=3	For ordinal state 0, bit 3. IF NODOPTS = 3, set to ON for a logical "1", OFF for a logical "0".

Continued on next page

8.2.2.12 Digital Output Point - 4-State Pulsed, continued

Table 8-20 4-State Pulsed Digital Output Parameter Entries, continued

Parameter	Entry	Required?	Notes
ST1_OP1	ON or OFF	Yes	For ordinal state 1, bit 1. Set to ON for a logical "1", OFF for a logical "0".
ST1_OP2	ON or OFF	Yes IF NODOPTS=2	For ordinal state 1, bit 2. IF NODOPTS = 2, set to ON for a logical "1", OFF for a logical "0".
ST1_OP3	ON or OFF	Yes IF NODOPTS=2	For ordinal state 1, bit 3. IF NODOPTS = 3, set to ON for a logical "1", OFF for a logical "0".
UNC_OP1	ON or OFF	Yes	For "stable" state, bit 1. Set to ON for a logical "1", OFF for a logical "0".
UNC_OP2	ON or OFF	Yes IF NODOPTS=2	For "stable" state, bit 2. IF NODOPTS = 2, set to ON for a logical "1", OFF for a logical "0".
UNC_OP3	ON or OFF	Yes IF NODOPTS=2	For "stable" state, bit 3. IF NODOPTS = 3, set to ON for a logical "1", OFF for a logical "0".
UNCLRTIM	-1, 0, or 1- <i>n</i>	Yes IF ALMUNCHG= ON	Configurable Clear of the Uncommanded Change alarm. IF ALMUNCHG = ON, set to: -1 for Operator clear only, 0 for automatic clear on next point execution 1- <i>n</i> for automatic clear after <i>n</i> seconds.

Continued on next page

8.2.2.12 Digital Output Point - 4-State Pulsed, continued

Build Example

A Digital point needs to be built to open and close valve #1. The valve requires output signal #1 = ON for OPEN and output signal #2 = ON for CLOSE. The output signals are pulsed for 1 second and return to the OFF state. There are no input signals to monitor the running state, so the output feedbacks must be checked to be sure that the command was accepted by the Digital device. The response time from the Digital device to set the commanded state is approximately 10 seconds. The operator needs to be able to command the valve and wants to know if the valve is opened or closed at the valve locally (uncommanded change). The interface is Modbus and therefore requires no additional CDS packages. Figure 8-18 below shows an example of an Exception Build file (in LCN R500 format) that could be used to create this image point.

Figure 8-18 Build Example for 4-State Pulsed Digital Output

```
{SCREEN ENTITY MOV_1( ) }
&T SWITCHAM
&N MOV_1
UNIT = C1
PTDESC ="VALVE 1 CONTROL  "
KEYWORD ="  "
CLSLOTS = 1
NOPKG = 3
NUMSWTCH = 2          {**NOTE: Will cause CL LINK ERROR if not set to 2**}
PRIMMOD = _____
SALMDSC1 ="  "
SALMDSC2 ="  "
SALMDSC3 ="  "
BCLEALPR = LOW        {**NOTE: R500 And Later**}
BCLFALPR = LOW        {**NOTE: R500 And Later**}
CLEALMPR = LOW        {**NOTE: R500 And Later**}
CLFALMPR = LOW        {**NOTE: R500 And Later**}
CNFERRPR = LOW        {**NOTE: R500 And Later**}
SWALM1PR = LOW        {**NOTE: R500 And Later**}
SWALM2PR = LOW        {**NOTE: R500 And Later**}
SWALM3PR = LOW        {**NOTE: R500 And Later**}
ALENBST = ENABLE
PKGNAME(1) ="CLM_DMD3"
PKGNAME(2) ="DIG4CDS "
PKGNAME(3) ="DIG4CDS1"
S1NSTATE = 4
S1STATES(0) ="OPEN  "  {**NOTE: User-entered text**}
S1STATES(1) ="CLOSE  " {**NOTE: User-entered text**}
S1STATES(2) ="MOVING "
S1STATES(3) ="BAD   "
S1ACCLVL = OPERATOR
```

Continued on next page

8.2.2.12 Digital Output Point - 4-State Pulsed, continued

Figure 8-18 Build Example for 4-State Pulsed Digital Output , continued

```
S1CURSTS = OPEN
S1REQSTS = OPEN
S2NSTATE = 2
S2STATES(0) = "CLEAR "
S2STATES(1) = "NOACTION"
S2ACCLVL = OPERATOR
S2CURSTS = NOACTION
S2REQSTS = NOACTION
$IPPASN = OFF
PERIOD = NOPERIOD          {**NOTE: Scheduled and driven by associated PIT**}
CLM_PORT = " "
CLM_DEV = " "
CLM_PIT = " "
NODINPTS = 0
NUMCMDST = 2
ALMOPT = 0
ALMUNCHG = ON
FBTIME = 20.10
TIMEOP = ON
NODOPTS = 2
ST0_OP1 = ON               {**NOTE: OPEN state sets output signal #1 to ON**}
ST0_OP2 = OFF
ST1_OP1 = OFF
ST1_OP2 = ON               {**NOTE: CLOSE state sets output signal #2 to ON**}
UNC_OP1 = OFF
UNC_OP2 = OFF
UNCLRTIM = -1
UNCCLEAR = OFF
DEV_STA = OFF
PT_FAIL = OFF
CMDCHG = OFF
ALMTYPE1 = A_NONE
ALMTYPE2 = A_NONE
ALMTYPE3 = A_NONE
INITLIZD = OFF
TIMER = 0
DATA(1) = 0
DATA(2) = 0
DATA(3) = 0
DATA(4) = 0
DATA(5) = 0
DATA(6) = 0
DATA(7) = 0
DATA(8) = 0
DATA(9) = 0
DATA(10) = 0
```

Continued on next page

8.2.2.12 Digital Output Point - 4-State Pulsed, continued

Attach the CL Block

All CLM Digital points require a CL Block. The 4-State Digital Output uses the DIG4CL.AO block included with the CLM Platform release media. Use the EC file included to link the block to the 4-State point.

From the COMMAND PROCESSOR window of the Engineering Personality:

EC DIG4.EC *pppppppp*

where

pppppppp is the 4-State point name

NOTE: 1) Be sure that your pathnames are set for the CL SOURCE/OBJ to point to the proper directory.
2) When executing this EC file for the first time, it is normal to get an error on the UNLINK since the block has not yet been attached.

Link Caution!

If the number of switches (NUMSWTCH) on the image point is not set to 2, the following error will be issued by the Linker.

Error during attempt to get parameter information; PA Status: END OF USER SPEC

Correct and rebuild the image point before attempting to re-link.

Activate the Point

At the Detail Display of the Digital point, set the PTEXECST = ACTIVE. You may need to PROCESS the point again to clear the RESTART state to NONE. The S1REQSTS and S1CURSTS should agree at this point, but if not, select the S2REQSTS, select the CLEAR option and press ENTER.

Map to the Device (See Application User Manual)

The build of the image is complete from the AM side. The next step is to assign the input and/or output signals to the required parameters. This is done through the CLM Configuration Builder and is application-specific. This means that each application will map to the same image point parameters, but the format and conversion process for each specific application will probably vary. Refer to the earlier sections on the CLM Configuration Builder in this manual and the specific application user manual for instructions on mapping your particular device.

Continued on next page

8.2.2.12 Digital Output Point - 4-State Pulsed, continued

Command the Point

The Digital point may command an output through the S1REQSTS parameter of the image point. *THE DIGITAL POINT MUST HAVE BEEN MAPPED ALREADY THROUGH THE CLM CONFIGURATION BUILDER.*

The steps to command an output are:

- Select the S1REQSTS parameter

- Select the desired state

- Press ENTER

This will PPS the point and cause the CL to execute, sending the appropriate OPn parameter values to the device.

<p>IMPORTANT! A mapped Digital point will not complete the S1REQSTS store if the associated device (NDT) is Out of Service or in the HARD-FAIL condition. This may be checked at the point by examining the DEV_STA parameter. If the value is OFF, then S1REQSTS stores will not complete and the operator will not see the requested change at the Detail Display.</p>

Clear the Point

After certain alarms or other conditions, it may become necessary to clear an existing alarm(s) and/or line up the S1REQSTS and S1CURSTS parameters. This can be done through the second switch on the Digital point. The steps to perform this action are:

- Select the S2REQSTS parameter

- Select the CLEAR state

- Press ENTER

This action will have the impact of setting the S1REQSTS = S1CURSTS, clearing all existing switch alarms, and re-issuing new alarms based on current conditions.

Operation

The 5 State Digital Composite point emulates a Motor-Operated-Valve (MOV) operation with 2 inputs. The input states reflect an OPEN/CLOSE condition (\$ord0-\$ord2), a STOP condition (STOP_ST), a MOVING or INBTWEEN condition (\$ord3), and a BAD or COMM_ERR condition (\$ord4). The D1 and D2 parameters reflect the current state of the Digital device, while the OP1, OP2 and OP3 parameters correspond to the latched output signals required to drive the Digital device.

CL is attached to the image point that:

- ⇒ Maps the D1 and D2 input signals to one of the ordinal input states (S1CURSTS).
- ⇒ When changed by the operator, maps the ordinal output state (S1REQSTS) to the OP1, OP2 and OP3 output signals, and executes the send to the Digital device.
- ⇒ When a STOP command is issued after an OPEN/CLOSE command, the current state is set to STOP if the S1REQSTS = STOP and the input signals D1 and D2 indicate the INBTWEEN state. From this point, either OPEN or CLOSE may be commanded.
- ⇒ When an uncommanded change to MOVING/INBTWEEN is detected, sets the feedback timer. If the state has not reached a limit value by the time that the timer has elapsed, the state is set to STOP and, if configured, an uncommanded change alarm is issued.
- ⇒ When an operator change is in progress, monitors the input signals (D1 and D2) for the requested state change against the configured time allowed for the transition (FBTIME), and either alarms or sets the ordinal state (S1CURSTS) to reflect the change.
- ⇒ If configured in ALMUNCHG, alarms when an input state change (D1 and D2) occurs that was not commanded by the operator.
- ⇒ If configured in the alarm options (ALMOPT), issues Off Normal alarms when the input state (D1 and D2) changes off of the normal state (PVNORMAL).

The FBTIME timer is actually two timers that monitor two conditions. When the operator change command is generated, the first timer (fractional part of $FBTIME * 100$) is set. If the input state has not changed to MOVING/INBTWEEN when this timer has elapsed, a NO_MOVE alarm is issued. If it has, the second timer (the difference between the whole part of FBTIME and the value of the first timer) is initiated. If the input state has not changed to the requested state when the second timer has elapsed, then a CMDDIS (command disagree) alarm is issued.

Alarms may be cleared or the requested state set equal to the current state by selecting the S2REQSTS parameter from the Detail Display for the point and selecting the CLEAR option.

Application

Digital device type examples:

Valve Control

Devices requiring 2 inputs and 3 latched output signals

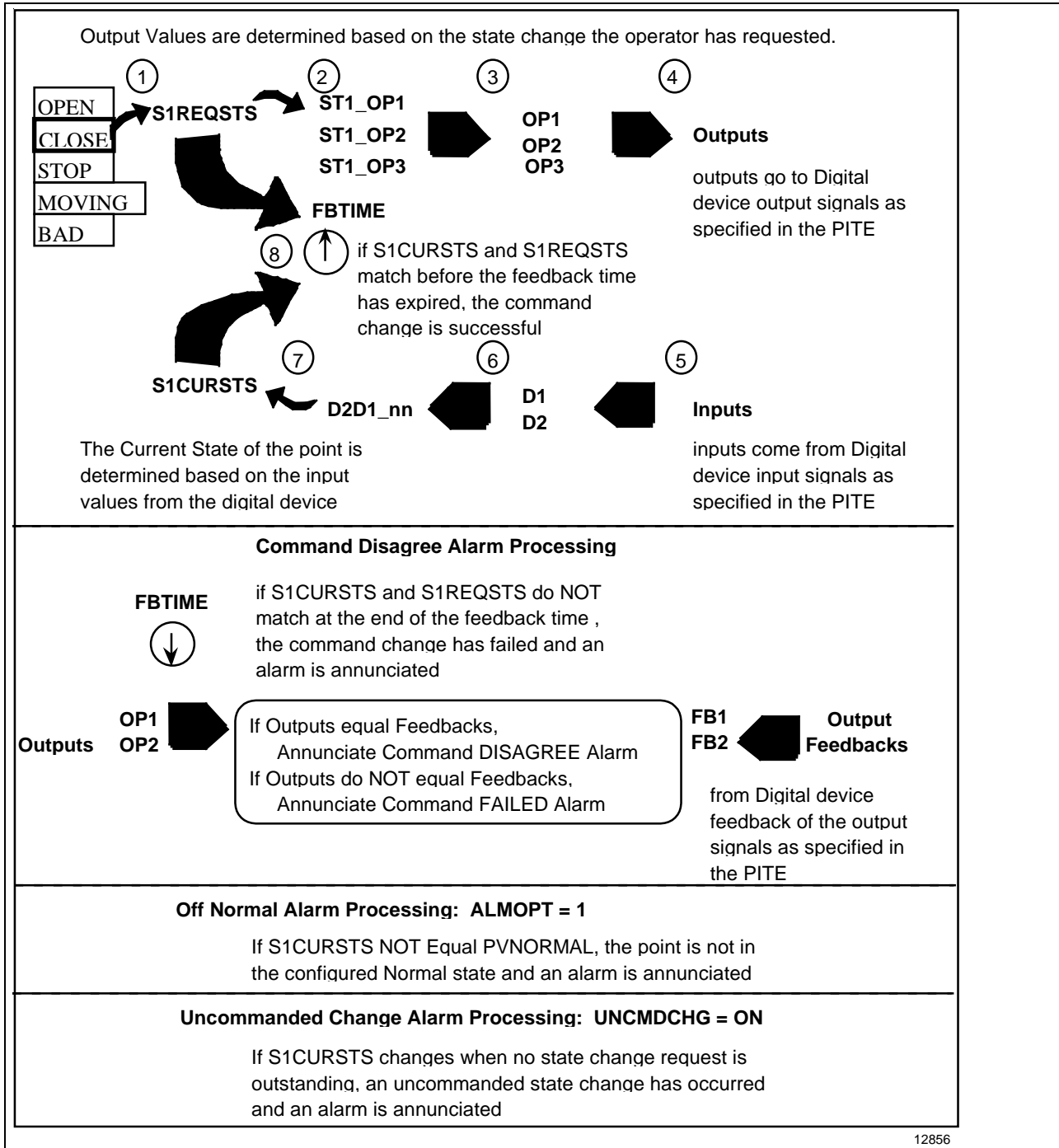
Continued on next page

8.2.2.13 Digital Composite Point, 5-State Latched, continued

Operational Flow

Following is a depiction of how a 5-State Digital Composite operates. It shows a Digital Composite with three output signals that are **Latched** Outputs with two input signals.

Figure 8-19 Operational Flow for 5-State Latched Digital Composite



Continued on next page

8.2.2.13 Digital Composite Point - 5-State Latched, continued

I/O Configuration Options

Number of Inputs	Number of Outputs	Max Cmd States
2	3	3

Alarming

The following is a list of alarms that may be issued by this Digital point type. Refer to Table 8-9 for specific alarm definition.

- OUT_SERV
 - PT_FAIL
 - CFG_ERR_*n*
 - INVCMD
 - CMDFAIL
 - CMDDIS
 - NO_MOVE
 - UNC_CHG (If configured through ALMUNCHG)
 - OFF_NORM (If ALMOPT = 1)
 - BAD_PV
 - BADSTATE
-

Build the Point

The 5-State Digital point is based on the AM Switch point. There are two methods of building this point type. Both of these methods use the Data Entity Builder (DEB) and the data entry requirements listed in Table 8-21 below.

- 1) Select the AM SWITCH point type and enter data as listed below.
or
- 2) Copy the DIG5L.EB Exception Build file (included and installed with the Platform release media) to another file for editing.

Edit the new Exception Build file using the data listed below as input for each of the points to be included within the file. Cut and Paste as required.

Perform an Exception Build process on the new file.

Refer to Table 8-10 for complete descriptions of all Digital point parameters.

See Figure 8-23 an Exception Build file example.

Continued on next page

8.2.2.13 Digital Composite Point - 5-State Latched, continued

Table 8-21 5-State Latched Digital Composite Point Parameter Entries

Parameter	Entry	Required?	Notes
NAME		Yes	See PKGNAME(4) notes below Application Specific - Refer to the User Manual section for any required packages
UNIT		Yes	
CLSLOTS	1	Yes	
NOPKG	3 or 4	Yes	
NUMSWTCH	2	Yes	
PKGNAME(1)	CLM_DMD3	Yes	
PKGNAME(2)	DIG5CDS	Yes	
PKGNAME(3)	DIG5CDS1	Yes	
PKGNAME(4)	See Notes	-	
S1NSTATE	5	Yes	Enter text for the ordinal 0 (or top) state Enter text for the ordinal 1 state Enter text for the ordinal 2 (normally stop) state Enter text for the ordinal 3 (between or moving) state Ordinal 4 (or bottom) state text. This state must be the BAD state. Key Access Level for command changes. Default is ENGINEER.
S1STATE(0)	See Notes	Yes	
S1STATE(1)	See Notes	Yes	
S1STATE(2)	See Notes	Yes	
S1STATE(3)	See Notes	Yes	
S1STATE(4)	BAD	Yes	
S1ACCLVL	Operator Supervisor Engineer Program	No	
S2NSTATE	2	Yes	Selected state for "resetting" the point. Default state. Selecting has no impact. Key Access Level for clearing switch alarms and setting commanded state = current state. This setting should match the S1ACCLVL. Default is ENGINEER.
S2STATE(0)	CLEAR	Yes	
S2STATE(1)	NOACTION	Yes	
S2ACCLVL	Operator Supervisor Engineer Program	No	
PERIOD	NOPERIOD	Yes	CLM Digital points are PPS'd for execution by data received by exception input or the CLM Point Interface Table (PIT) scheduler.

Continued on next page

8.2.2.13 Digital Composite Point - 5-State Latched, continued

Table 8-21 5-State Latched Digital Composite Point Parameter Requirements, continued

Parameter	Entry	Required?	Notes
D1_0	\$ord0-\$ord1	Yes	S1CURSTS becomes the ordinal state identified when D1 is OFF.
D1_1	\$ord0-\$ord1	Yes	S1CURSTS becomes the ordinal state identified when D1 is ON.
D2D1_01	\$ord0-\$ord3	Yes	Enter the ordinal state that maps to the condition where bit 1 (D1)=ON and bit 2 (D2)=OFF
D2D1_10	\$ord0-\$ord3	Yes	Enter the ordinal state that maps to the condition where bit 1 (D1)=OFF and bit 2 (D2)=ON
D2D1_11	\$ord0-\$ord3	Yes	Enter the ordinal state that maps to the condition where bit 1 (D1)=ON and bit 2 (D2)=ON
D2D1_00	\$ord0-\$ord3	Yes	Enter the ordinal state that maps to the condition where bit 1 (D1)=OFF and bit 2 (D2)=OFF
NUMCMDST	3	Yes	Set to 3 in all cases. OPEN, CLOSED, AND STOP used in the 5-state
ALMOPT	0-1	No	Default = 0 for operational alarms only. Set to 1 to add Off Normal alarming.
ALMUNCHG	ON or OFF	No	Default = OFF. Set to ON to alarm uncommanded state changes.
PVNORMAL	\$ord0-\$ord1	Yes IF ALMOPT=1	Not required when ALMOPT = 0. Required when ALMOPT = 1.
NODINPTS	2	Yes	Number of Digital Input signals is 2 for MOV operation.
FBTIME	See Notes TT.tt	Yes	Feedback Timeouts. Maximum times (in secs) allowed for the commanded state changes, where TT=total time, tt=no movement time. Refer to Table 8-10.
TIMEOP	OFF	Yes	Set to OFF for "Latched" outputs.
NODOPTS	3	Yes	Number of Digital output signals.
ST0_OP1	ON or OFF	Yes	For ordinal state 0, bit 1. Set to ON for a logical "1", OFF for a logical "0".
ST0_OP2	ON or OFF	Yes	For ordinal state 0, bit 2. IF NODOPTS = 2, set to ON for a logical "1", OFF for a logical "0".
ST0_OP3	ON or OFF	Yes	For ordinal state 0, bit 3. IF NODOPTS = 3, set to ON for a logical "1", OFF for a logical "0".

Continued on next page

8.2.2.13 Digital Composite Point - 5-State Latched, continued

Table 8-21 5-State Latched Digital Composite Point Parameter Requirements, continued

Parameter	Entry	Required?	Notes
ST1_OP1	ON or OFF	Yes	For ordinal state 1, bit 1. Set to ON for a logical "1", OFF for a logical "0".
ST1_OP2	ON or OFF	Yes	For ordinal state 1, bit 2. IF NODOPTS = 2, set to ON for a logical "1", OFF for a logical "0".
ST1_OP3	ON or OFF	Yes	For ordinal state 1, bit 3. IF NODOPTS = 3, set to ON for a logical "1", OFF for a logical "0".
ST2_OP1	ON or OFF	Yes	For ordinal state 2, bit 1. Set to ON for a logical "1", OFF for a logical "0".
ST2_OP2	ON or OFF	Yes	For ordinal state 2, bit 2. IF NODOPTS = 2, set to ON for a logical "1", OFF for a logical "0".
ST2_OP3	ON or OFF	Yes	For ordinal state 2, bit 3. IF NODOPTS = 3, set to ON for a logical "1", OFF for a logical "0".
STOP_ST	0-2	Yes	Ordinal state assigned to the STOP command and position.
MOVE_ST	3	Yes	Ordinal state assigned to the MOVING or INBTWEEN position.
UNCLRTIM	-1, 0, or 1- <i>n</i>	Yes IF ALMUNCHG= ON	Configurable Clear of the Uncommanded Change alarm. IF ALMUNCHG = ON, set to: -1 for Operator clear only, 0 for automatic clear on next point execution 1- <i>n</i> for automatic clear after <i>n</i> seconds.

Continued on next page

8.2.2.13 Digital Composite Point - 5-State Latched, continued

Build Example

The user wants to build a Digital point to emulate a valve with stop capability. The valve requires output signal #1 = ON for CLOSE, output signal #2 = ON for OPEN, and output signal #3 = ON for STOP. The running state of the valve is monitored through the two limit switches. Input signal #1 (D1) from the CLOSED limit switch and input signal #2 (D2) from the OPENED limit switch. The limit switches are normally open, so the input signals reflect OFF and OFF when the valve is MOVING or INBTWEEN stops. It takes approximately 45 seconds for the valve to completely transition from one stop to the other. If the valve has not started moving after 15 seconds, then there is a problem or a jam. The operator needs to be able to command the valve and it cannot be started or stopped at the valve (no uncommanded change). The interface is through a Modbus interface to a PLC and therefore no additional CDS packages are required. Figure 8-20 below shows an example of an Exception Build file (in LCN R500 format) that could be used to create this image point.

Figure 8-20 Build Example for 5-State Latched Digital Composite

```
{SCREEN ENTITY VALVE_1( ) }
&T SWITCHAM
&N VALVE_1
UNIT = C1
PTDESC ="VALVE 1 CONTROL  "
KEYWORD ="  "
CLSLOTS = 1
NOPKG = 3
NUMSWTCH = 2          {**NOTE: Will cause CL LINK ERROR if not set to 2**}
PRIMMOD = _____
SALMDSC1 ="  "
SALMDSC2 ="  "
SALMDSC3 ="  "
BCLEALPR = LOW        {**NOTE: R500 And Later**}
BCLFALPR = LOW        {**NOTE: R500 And Later**}
CLEALMPR = LOW        {**NOTE: R500 And Later**}
CLFALMPR = LOW        {**NOTE: R500 And Later**}
CNFERRPR = LOW        {**NOTE: R500 And Later**}
SWALM1PR = LOW        {**NOTE: R500 And Later**}
SWALM2PR = LOW        {**NOTE: R500 And Later**}
SWALM3PR = LOW        {**NOTE: R500 And Later**}
ALENBST = ENABLE
PKGNAME(1) ="CLM_DMD3"
PKGNAME(2) ="DIG5CDS "
PKGNAME(3) ="DIG5CDS1"
SINSTATE = 5
```

Continued on next page

8.2.2.13 Digital Composite Point - 5-State Latched, continued

Figure 8-20 Build Example for 5-State Latched Digital Composite , continued

```
S1STATES(0) ="OPEN " {**NOTE: User-entered text**}
S1STATES(1) ="CLOSE"      {**NOTE: User-entered text**}
S1STATES(2) ="STOP  "
S1STATES(3) ="MOVING "
S1STATES(4) ="BAD  "
S1ACCLVL = OPERATOR
S1CURSTS = CLOSE
S1REQSTS = CLOSE
S2NSTATE = 2
S2STATES(0) ="CLEAR  "
S2STATES(1) ="NOACTION"
S2ACCLVL = OPERATOR
S2CURSTS = NOACTION
S2REQSTS = NOACTION
$IPPASN = OFF
PERIOD  = NOPERIOD      {**NOTE: Scheduled and driven by associated PIT**}
CLM_PORT ="  "
CLM_DEV  ="  "
CLM_PIT  ="  "
NODINPTS = 2
D1_0  = $ORD0      {**NOTE: Valve is OPEN when state is logical zero**}
D1_1  = $ORD1
D2D1_01 = $ORD1      {**CLOSE**}
D2D1_10 = $ORD0      {**OPEN**}
D2D1_00 = $ORD2      {**MOVING or STOP**}
D2D1_11 = $ORD3      {**BAD**}
NUMCMDST = 3
ALMOPT  = 0
ALMUNCHG = OFF
PVNORMAL = $ORD1
FBTIME  = 45.15
TIMEOP  = OFF      {**NOTE: MUST be OFF for Latched output signals**}
NODOPTS = 3
ST0_OP1 = OFF      {**NOTE: OPEN state sets output signal #2 to ON**}
ST0_OP2 = ON
ST0_OP3 = OFF
ST1_OP1 = ON      {**NOTE: CLOSE state sets output signal #1 to ON**}
ST1_OP2 = OFF
ST1_OP3 = OFF
ST2_OP1 = ON      {**NOTE: STOP state sets output signal #3 to ON**}
ST2_OP2 = OFF
ST2_OP3 = OFF
```

Continued on next page

8.2.2.13 Digital Composite Point - 5-State Latched, continued

Figure 8-20 Build Example for 5-State Latched Digital Composite , continued

```
STOP_ST = 2
MOVE_ST = 3
UNCLRTIM = -1
UNCCLEAR = OFF
DEV_STA = OFF
PT_FAIL = OFF
CMDCHG = OFF
ALMTYPE1 = A_NONE
ALMTYPE2 = A_NONE
ALMTYPE3 = A_NONE
INITLIZD = OFF
TIMER = 0
DATA(1) = 0
DATA(2) = 0
DATA(3) = 0
DATA(4) = 0
DATA(5) = 0
DATA(6) = 0
DATA(7) = 0
DATA(8) = 0
DATA(9) = 0
DATA(10) = 0
```

Continued on next page

8.2.2.13 Digital Composite Point - 5-State Latched, continued

Attach the CL Block

All CLM Digital points require a CL Block. The 5-State Digital Composite uses the DIG5CL.AO block included with the CLM Platform release media. Use the EC file included to link the block to the 5-State point.

From the COMMAND PROCESSOR window of the Engineering Personality:

```
EC DIG5.EC pppppppp
      where
      pppppppp is the 5-State point name
```

NOTE: 1) Be sure that your pathnames are set for the CL SOURCE/OBJ to point to the proper directory.
2) When executing this EC file for the first time, it is normal to get an error on the UNLINK since the block has not yet been attached.

Link Caution!

If the number of switches (NUMSWTCH) on the image point is not set to 2, the following error will be issued by the Linker.

Error during attempt to get parameter information; PA Status: END OF USER SPEC

Correct and rebuild the image point before attempting to re-link.

Activate the Point

At the Detail Display of the Digital point, set the PTEXECST = ACTIVE. You may need to PROCESS the point again to clear the RESTART state to None. The S1REQSTS and S1CURSTS should agree at this point, but if not, select the S2REQSTS, select the CLEAR option and press ENTER.

Map to the Device (See Application User Manual)

The build of the image is complete from the AM side. The next step is to assign the input and/or output signals to the required parameters. This is done through the CLM Configuration Builder and is application-specific. This means that each application will map to the same image point parameters, but the format and conversion process for each specific application will probably vary. Refer to the earlier sections on the CLM Configuration Builder in this manual and the specific application user manual for instructions on mapping your particular device.

Continued on next page

8.2.2.13 Digital Composite Point - 5-State Latched, continued

Command the Point

The Digital point may command an output through the S1REQSTS parameter of the image point. *THE DIGITAL POINT MUST HAVE BEEN MAPPED ALREADY THROUGH THE CLM CONFIGURATION BUILDER.*

The steps to command an output are:

- Select the S1REQSTS parameter

- Select the desired state

- Press ENTER

This will PPS the point and cause the CL to execute, sending the appropriate OPn parameter values to the device.

<p>IMPORTANT! A mapped Digital point will not complete the S1REQSTS store if the associated device (NDT) is Out of Service or in the HARD-FAIL condition. This may be checked at the point by examining the DEV_STA parameter. If the value is OFF, then S1REQSTS stores will not complete and the operator will not see the requested change at the Detail Display.</p>

Clear the Point

After certain alarms or other conditions, it may become necessary to clear an existing alarm(s) and/or line up the S1REQSTS and S1CURSTS parameters. This can be done through the second switch on the Digital point. The steps to perform this action are:

- Select the S2REQSTS parameter

- Select the CLEAR state

- Press ENTER

This action will have the impact of setting the S1REQSTS = S1CURSTS, clearing all existing switch alarms, and re-issuing new alarms based on current conditions.

Operation

The 5 State Digital Composite point emulates a Motor-Operated-Valve (MOV) operation with 2 inputs. The input states reflect an OPEN/CLOSE condition (\$ord0-\$ord2), a STOP condition (STOP_ST), a MOVING or INBTWEEN condition (\$ord3), and a BAD or COMM_ERR condition (\$ord4). The D1 and D2 parameters reflect the current state of the Digital device, while the OP1, OP2 and OP3 parameters correspond to the pulsed output signals required to drive the Digital device.

CL is attached to the image point that:

- ⇒ Maps the D1 and D2 input signals to one of the ordinal input states (S1CURSTS).
- ⇒ When changed by the operator, maps the ordinal output state (S1REQSTS) to the OP1, OP2 and OP3 output signals, and executes the send to the Digital device.
- ⇒ When a STOP command is issued after an OPEN/CLOSE command, the current state is set to STOP if the S1REQSTS = STOP and the input signals D1 and D2 indicate the INBTWEEN state. From this point, either OPEN or CLOSE may be commanded.
- ⇒ When an uncommanded change to MOVING/INBTWEEN is detected, sets the feedback timer. If the state has not reached a limit value by the time that the timer has elapsed, the state is set to STOP and, if configured, an uncommanded change alarm is issued.
- ⇒ When an operator change is in progress, monitors the input signals (D1 and D2) for the requested state change against the configured time allowed for the transition (FBTIME), and either alarms or sets the ordinal state (S1CURSTS) to reflect the change.
- ⇒ If configured in ALMUNCHG, alarms when an input state change (D1 and D2) occurs that was not commanded by the operator.

The FBTIME timer is actually two timers that monitor two conditions. When the operator change command is generated, the first timer (fractional part of FBTIME * 100) is set. If the input state has not changed to MOVING/INBTWEEN when this timer has elapsed, a NO_MOVE alarm is issued. If it has, the second timer (the difference between the whole part of FBTIME and the value of the first timer) is initiated. If the input state has not changed to the requested state when the second timer has elapsed, then a CMDDIS (command disagree) alarm is issued.

Alarms may be cleared or the requested state set equal to the current state by selecting the S2REQSTS parameter from the Detail Display for the point and selecting the CLEAR option.

Application

Digital device type examples:

Valve control

Devices requiring 2 inputs and 3 pulsed output signals

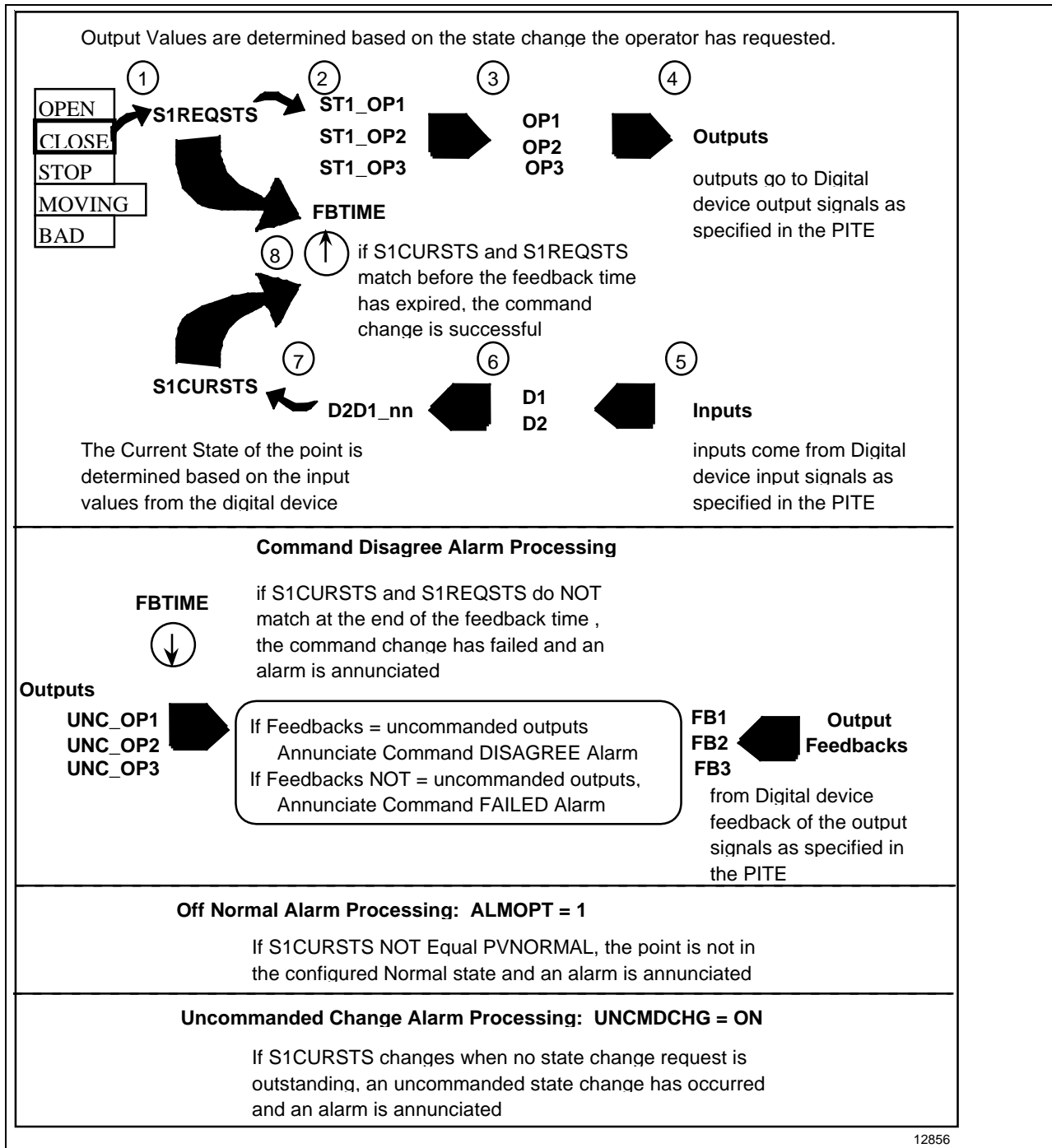
Continued on next page

8.2.2.14 Digital Composite Point - 5-State Pulsed, continued

Operational Flow

Following is a depiction of how a 5-State Digital Composite operates. It shows a Digital Composite with two output signals that are **Pulsed** Outputs with two input signals.

Figure 8-21 Operational Flow for 5-State Pulsed Digital Composite



Continued on next page

8.2.2.14 Digital Composite Point - 5-State Pulsed, continued

I/O Configuration Options

Number of Inputs	Number of Outputs	Max Cmd States
2	3	3

Alarming

The following is a list of alarms that may be issued by this Digital point type. Refer to Table 8-9 for specific alarm definition.

- OUT_SERV
- PT_FAIL
- CFG_ERR_*n*
- INVCMD
- CMDFAIL
- CMDDIS
- NO_MOVE
- UNC_CHG (If configured through ALMUNCHG)
- OFF_NORM (If ALMOPT = 1)
- BAD_PV
- BADSTATE

Build the Point

The 5-State Digital point is based on the AM Switch point. There are two methods of building this point type. Both of these methods use the Data Entity Builder (DEB) and the data entry requirements listed in Table 8-22 below.

- 1) Select the AM SWITCH point type and enter data as listed below.

Or

- 2) Copy the DIG5P.EB Exception Build file (included and installed with the Platform release media) to another file for editing.

Edit the new Exception Build file using the data listed below as input for each of the points to be included within the file. Cut and Paste as required.

Perform an Exception Build process on the new file.

Refer to Table 8-10 for complete descriptions of all Digital point parameters.

See Figure 8-22 for an Exception Build file example.

Continued on next page

8.2.2.14 Digital Composite Point - 5-State Pulsed, continued

Table 8-22 5-State Pulsed Digital Composite Point Parameter Entries

Parameter	Entry	Required?	Notes
NAME		Yes	See PKGNAME(4) notes below
UNIT		Yes	
CLSLOTS	1	Yes	
NOPKG	3 or 4	Yes	
NUMSWTCH	2	Yes	
PKGNAME(1)	CLM_DMD3	Yes	
PKGNAME(2)	DIG5CDS	Yes	
PKGNAME(3)	DIG5CDS1	Yes	
PKGNAME(4)	See Notes	-	
S1NSTATE	5	Yes	<p>Enter text for the ordinal 0 (or top) state</p> <p>Enter text for the ordinal 1 state</p> <p>Enter text for the ordinal 2 (normally stop) state</p> <p>Enter text for the ordinal 3 (between or moving) state</p> <p>Ordinal 4 (or bottom) state text. This state must be the BAD state.</p> <p>Key Access Level for command changes. Default is ENGINEER.</p>
S1STATE(0)	See Notes	Yes	
S1STATE(1)	See Notes	Yes	
S1STATE(2)	See Notes	Yes	
S1STATE(3)	See Notes	Yes	
S1STATE(4)	BAD	Yes	
S1ACCLVL	Operator Supervisor Engineer Program	No	
S2NSTATE	2	Yes	<p>Selected state for “resetting” the point.</p> <p>Default state. Selecting has no impact.</p> <p>Key Access Level for clearing switch alarms and setting commanded state = current state. This setting should match the S1ACCLVL. Default is ENGINEER.</p>
S2STATE(0)	CLEAR	Yes	
S2STATE(1)	NOACTION	Yes	
S2ACCLVL	Operator Supervisor Engineer Program	No	
PERIOD	NOPERIOD	Yes	CLM Digital points are PPS'd for execution by data received by exception input or the CLM Point Interface Table (PIT) scheduler.

Continued on next page

8.2.2.14 Digital Composite Point - 5-State Pulsed, continued

Table 8-22 5-State Pulsed Digital Composite Point Parameter Requirements, continued

Parameter	Entry	Required?	Notes
D1_0	\$ord0-\$ord1	Yes	S1CURSTS becomes the ordinal state identified when D1 is OFF.
D1_1	\$ord0-\$ord1	Yes	S1CURSTS becomes the ordinal state identified when D1 is ON.
D2D1_01	\$ord0-\$ord3	Yes	Enter the ordinal state that maps to the condition where bit 1 (D1)=ON and bit 2 (D2)=OFF
D2D1_10	\$ord0-\$ord3	Yes	Enter the ordinal state that maps to the condition where bit 1 (D1)=OFF and bit 2 (D2)=ON
D2D1_11	\$ord0-\$ord3	Yes	Enter the ordinal state that maps to the condition where bit 1 (D1)=ON and bit 2 (D2)=ON
D2D1_00	\$ord0-\$ord3	Yes	Enter the ordinal state that maps to the condition where bit 1 (D1)=OFF and bit 2 (D2)=OFF
NUMCMDST	1-2	Yes	Set to 1 if only ordinal state 0 can be commanded. Set to 2 if both ordinal 0 and 1 can be commanded.
ALMOPT	0-1	No	Default = 0 for operational alarms only. Set to 1 to add Off Normal alarming.
ALMUNCHG	ON or OFF	No	Default = OFF. Set to ON to alarm uncommanded state changes.
PVNORMAL	\$ord0-\$ord1	Yes IF ALMOPT=1	Not required when ALMOPT = 0. Required when ALMOPT = 1.
NODINPTS	2	Yes	Number of Digital Input signals is 2.
FBTIME	See Notes TT.tt	Yes	Feedback Timeouts. Maximum times (in secs) allowed for the commanded state changes, where TT=total time, tt=no movement time. Refer to Table 8-10.
TIMEOP	ON	Yes	Set to ON for "Pulsed" outputs.
NODOPTS	3	Yes	Number of Digital output signals.
ST0_OP1	ON or OFF	Yes	For ordinal state 0, bit 1. Set to ON for a logical "1", OFF for a logical "0".
ST0_OP2	ON or OFF	Yes	For ordinal state 0, bit 2. IF NODOPTS = 2, set to ON for a logical "1", OFF for a logical "0".
ST0_OP3	ON or OFF	Yes	For ordinal state 0, bit 3. IF NODOPTS = 3, set to ON for a logical "1", OFF for a logical "0".

Continued on next page

8.2.2.14 Digital Composite Point - 5-State Pulsed, continued

Table 8-22 5-State Pulsed Digital Composite Point Parameter Requirements, continued

Parameter	Entry	Required?	Notes
ST1_OP1	ON or OFF	Yes	For ordinal state 1, bit 1. Set to ON for a logical "1", OFF for a logical "0".
ST1_OP2	ON or OFF	Yes	For ordinal state 1, bit 2. IF NODOPTS = 2, set to ON for a logical "1", OFF for a logical "0".
ST1_OP3	ON or OFF	Yes	For ordinal state 1, bit 3. IF NODOPTS = 3, set to ON for a logical "1", OFF for a logical "0".
ST2_OP1	ON or OFF	Yes	For ordinal state 2, bit 1. Set to ON for a logical "1", OFF for a logical "0".
ST2_OP2	ON or OFF	Yes	For ordinal state 2, bit 2. IF NODOPTS = 2, set to ON for a logical "1", OFF for a logical "0".
ST2_OP3	ON or OFF	Yes	For ordinal state 2, bit 3. IF NODOPTS = 3, set to ON for a logical "1", OFF for a logical "0".
UNC_OP1	ON or OFF	Yes	For output bit 1, this is the stable, or non-pulsed, state. Set to On for a logical "1", OFF for a logical "0".
UNC_OP2	ON or OFF	Yes	For output bit 2, this is the stable, or non-pulsed, state. Set to On for a logical "1", OFF for a logical "0".
UNC_OP3	ON or OFF	Yes	For output bit 3, this is the stable, or non-pulsed, state. Set to On for a logical "1", OFF for a logical "0".
STOP_ST	0-2	Yes	Ordinal state assigned to the STOP command and position.
MOVE_ST	3	Yes	Ordinal state assigned to the MOVING or INBTWEEN position.
UNCLRTIM	-1, 0, or 1- <i>n</i>	Yes IF ALMUNCHG= ON	Configurable Clear of the Uncommanded Change alarm. IF ALMUNCHG = ON, set to: -1 for Operator clear only, 0 for automatic clear on next point execution 1- <i>n</i> for automatic clear after <i>n</i> seconds.

Continued on next page

8.2.2.14 Digital Composite Point - 5-State Pulsed, continued

Build Example

The user wants to build a Digital point to emulate a valve with stop capability. The valve requires a 4 second ON to output signal #1 for an OPEN command, a 4 second ON to output signal #2 for a CLOSE command, and a 4 second ON to output signal #3 for a STOP command. The resting, or return state for all output states is OFF. The running state of the valve is monitored through the two limit switches. Input signal #1 (D1) from the CLOSE limit switch and input signal #2 (D2) from the OPEN limit switch. The limit switches are normally open, so the input signals reflect OFF and OFF when the valve is MOVING or INBTWEEN stops. It takes approximately 30 seconds for the valve to completely transition from one stop to the other. If the valve has not started moving after 10 seconds, then there is a problem or a jam. The operator needs to be able to command the valve and it can be started or stopped locally at the valve (uncommanded change). The interface is through a Modbus interface to a PLC and therefore no additional CDS packages are required. Figure 8-22 below shows an example of an Exception Build file (in LCN R500 format) that could be used to create this image point.

Figure 8-22 Build Example for 5-State Pulsed Digital Composite

```
{SCREEN ENTITY VALVE_1() }
&T SWITCHAM
&N VALVE_1
UNIT = C1
PTDESC ="VALVE 1 CONTROL "
KEYWORD =" "
CLSLOTS = 1
NOPKG = 3
NUMSWTCH = 2 {**NOTE: Will cause CL LINK ERROR if not set to 2**}
PRIMMOD = _____
SALMDSC1 =" "
SALMDSC2 =" "
SALMDSC3 =" "
BCLEALPR = LOW {**NOTE: R500 And Later**}
BCLFALPR = LOW {**NOTE: R500 And Later**}
CLEALMPR = LOW {**NOTE: R500 And Later**}
CLFALMPR = LOW {**NOTE: R500 And Later**}
CNFERRPR = LOW {**NOTE: R500 And Later**}
SWALM1PR = LOW {**NOTE: R500 And Later**}
SWALM2PR = LOW {**NOTE: R500 And Later**}
SWALM3PR = LOW {**NOTE: R500 And Later**}
ALENBST = ENABLE
PKGNAME(1) ="CLM_DMD3"
PKGNAME(2) ="DIG5CDS "
PKGNAME(3) ="DIG5CDS1"
```

Continued on next page

8.2.2.14 Digital Composite Point - 5-State Pulsed, continued

Figure 8-22 Build Example for 5-State Pulsed Digital Composite , continued

```
S1NSTATE = 4
S1STATES(0) = "OPEN " {**NOTE: User-entered text**}
S1STATES(1) = "CLOSE" {**NOTE: User-entered text**}
S1STATES(2) = "STOP "
S1STATES(3) = "MOVING "
S1STATES(4) = "BAD "
S1ACCLVL = OPERATOR
S1CURSTS = CLOSE
S1REQSTS = CLOSE
S2NSTATE = 2
S2STATES(0) = "CLEAR "
S2STATES(1) = "NOACTION"
S2ACCLVL = OPERATOR
S2CURSTS = NOACTION
S2REQSTS = NOACTION
$IPPASN = OFF
PERIOD = NOPERIOD {**NOTE: Scheduled and driven by associated PIT**}
CLM_PORT = " "
CLM_DEV = " "
CLM_PIT = " "
NODINPTS = 2
D1_0 = $ORD0 {**NOTE: Valve is OPEN when state is logical zero**}
D1_1 = $ORD1
D2D1_01 = $ORD1 {**CLOSE**}
D2D1_10 = $ORD0 {**OPEN**}
D2D1_00 = $ORD2 {**MOVING or STOP**}
D2D1_11 = $ORD3 {**BAD**}
NUMCMDST = 3
ALMOPT = 0
ALMUNCHG = ON
PVNORMAL = $ORD1
FBTIME = 30.10
TIMEOP = ON {**NOTE: MUST be ON for Pulsed output signals**}
NODOPTS = 2
ST0_OP1 = OFF {**NOTE: OPEN state sets output signal #2 to ON**}
ST0_OP2 = ON
ST0_OP3 = OFF
ST1_OP1 = ON {**NOTE: CLOSE state sets output signal #1 to ON**}
ST1_OP2 = OFF
ST1_OP3 = OFF
ST2_OP1 = ON {**NOTE: STOP state sets output signal #3 to ON**}
ST2_OP2 = OFF
ST2_OP3 = OFF
UNC_OP1 = OFF
UNC_OP2 = OFF
UNC_OP3 = OFF
```

Continued on next page

8.2.2.14 Digital Composite Point - 5-State Pulsed, continued

Figure 8-22 Build Example for 5-State Pulsed Digital Composite , continued

```
STOP_ST = 2
MOVE_ST = 3
UNCLRTIM = -1
UNCCLEAR = OFF
DEV_STA = OFF
PT_FAIL = OFF
CMDCHG = OFF
ALMTYPE1 = A_NONE
ALMTYPE2 = A_NONE
ALMTYPE3 = A_NONE
INITLIZD = OFF
TIMER = 0
DATA(1) = 0
DATA(2) = 0
DATA(3) = 0
DATA(4) = 0
DATA(5) = 0
DATA(6) = 0
DATA(7) = 0
DATA(8) = 0
DATA(9) = 0
DATA(10) = 0
```

Continued on next page

8.2.2.14 Digital Composite Point - 5-State Pulsed, continued

Attach the CL Block

All CLM Digital points require a CL Block. The 5-State Digital Composite uses the DIG5CL.AO block included with the CLM Platform release media. Use the EC file included to link the block to the 5-State point.

From the COMMAND PROCESSOR window of the Engineering Personality:

```
EC DIG5P.EC pppppppp
where
pppppppp is the 5-State point name
```

NOTE: 1) Be sure that your pathnames are set for the CL SOURCE/OBJ to point to the proper directory.
2) When executing this EC file for the first time, it is normal to get an error on the UNLINK since the block has not yet been attached.

Link Caution!

If the number of switches (NUMSWTCH) on the image point is not set to 2, the following error will be issued by the Linker.

Error during attempt to get parameter information; PA Status: END OF USER SPEC

Correct and rebuild the image point before attempting to re-link.

Activate the Point

At the Detail Display of the Digital point, set the PTEXECST = ACTIVE. You may need to PROCESS the point again to clear the RESTART state to None. The S1REQSTS and S1CURSTS should agree at this point, but if not, select the S2REQSTS, select the CLEAR option and press ENTER.

Map to the Device (See Application User Manual)

The build of the image is complete from the AM side. The next step is to assign the input and/or output signals to the required parameters. This is done through the CLM Configuration Builder and is application-specific. This means that each application will map to the same image point parameters, but the format and conversion process for each specific application will probably vary. Refer to the earlier sections on the CLM Configuration Builder in this manual and the specific application user manual for instructions on mapping your particular device.

Continued on next page

8.2.2.14 Digital Composite Point - 5-State Pulsed, continued

Command the Point

The Digital point may command an output through the S1REQSTS parameter of the image point. *THE DIGITAL POINT MUST HAVE BEEN MAPPED ALREADY THROUGH THE CLM CONFIGURATION BUILDER.*

The steps to command an output are:

Select the S1REQSTS parameter

Select the desired state

Press ENTER

This will PPS the point and cause the CL to execute, sending the appropriate OPn parameter values to the device.

<p>IMPORTANT! A mapped Digital point will not complete the S1REQSTS store if the associated device (NDT) is Out of Service or in the HARD-FAIL condition. This may be checked at the point by examining the DEV_STA parameter. If the value is OFF, then S1REQSTS stores will not complete and the operator will not see the requested change at the Detail Display.</p>

Clear the Point

After certain alarms or other conditions, it may become necessary to clear an existing alarm(s) and/or line up the S1REQSTS and S1CURSTS parameters. This can be done through the second switch on the Digital point. The steps to perform this action are:

Select the S2REQSTS parameter

Select the CLEAR state

Press ENTER

This action will have the impact of setting the S1REQSTS = S1CURSTS, clearing all existing switch alarms, and re-issuing new alarms based on current conditions.

Operation

The 5 State Digital Output point emulates a Motor-Operated-Valve (MOV) operation with no inputs. The feedback states reflect an OPEN/CLOSE condition (\$ord0-\$ord2), a STOP condition (STOP_ST), a MOVING or INBTWEEN condition (\$ord3), and a BAD or COMM_ERR condition (\$ord4). The D1 and D2 are not used since there are no separate input signals, the OP1, OP2 and OP3 parameters correspond to the latched output signals required to drive the Digital device, and the FB1, FB2, and FB3 parameters correspond to the feedback signals required for output verification.

When a STOP command is issued after an OPEN/CLOSE command, the current state will be set to STOP if the S1REQSTS = STOP and the input signals D1 and D2 indicate the INBTWEEN state. From this point, either OPEN or CLOSE may be commanded. Uncommanded change for the STOP command is supported by the CL. Since no indicator for the STOP is returned in most cases, an uncommanded change to MOVING/INBTWEEN will set the feedback timer. If the state has not reached a limit value by the time that the timer has elapsed, the state is set to STOP and, if configured, an uncommanded change alarm is issued.

The FBTIME timer is actually two timers that monitor two conditions. When the operator change command is generated, the first timer (fractional part of FBTIME * 100) is set. If the input state has not changed to MOVING/INBTWEEN when this timer has elapsed, a NO_MOVE alarm is issued. If it has, the second timer (the difference between the whole part of FBTIME and the value of the first timer) is initiated. If the input state has not changed to the requested state when the second timer has elapsed, then a CMDDIS (command disagree) alarm is issued.

Alarms may be cleared or the requested state set equal to the current state by selecting the S2REQSTS parameter from the Detail Display for the point and selecting the CLEAR option.

Application

Digital device type examples:

- Pump motor monitor and control
- Valve control
- Devices requiring 0 inputs and 3 latched output signals

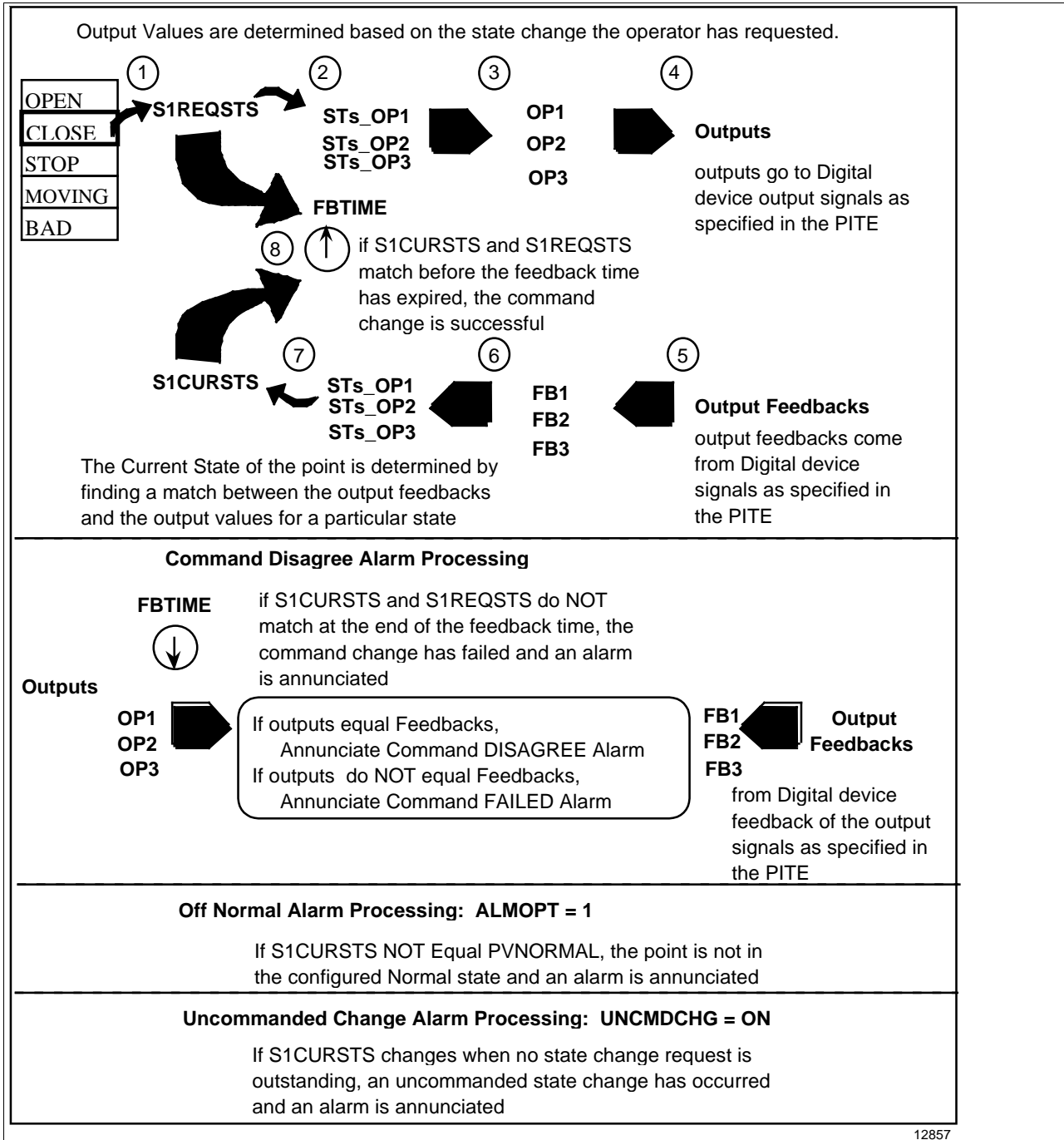
Continued on next page

8.2.2.15 Digital Output Point, 5-State Latched, continued

Operational Flow

Following is a depiction of how a 5-State Digital Output operates. It shows a Digital Output with three output signals that are **Latched** Outputs and no input signals.

Figure 8-23 Operational Flow for 5-State Latched Digital Output



Continued on next page

8.2.2.15 Digital Output Point - 5-State Latched, continued

I/O Configuration Options

Number of Inputs	Number of Outputs	Max Cmd States
0	3	3

Alarming

The following is a list of alarms that may be issued by this Digital point type. Refer to Table 8-9 for specific alarm definition.

- OUT_SERV
- PT_FAIL
- CFG_ERR_*n*
- INVCMD
- CMDFAIL
- CMDDIS
- NO_MOVE
- UNC_CHG (If configured through ALMUNCHG)
- BAD_PV
- BADSTATE

Build the Point

The 5-State Digital point is based on the AM Switch point. There are two methods of building this point type. Both of these methods use the Data Entity Builder (DEB) and the data entry requirements listed in Table 8-23 below.

- 1) Select the AM SWITCH point type and enter data as listed below.
or
- 2) Copy the DOUT5L.EB Exception Build file (included and installed with the Platform release media) to another file for editing.

Edit the new Exception Build file using the data listed below as input for each of the points to be included within the file. Cut and Paste as required.

Perform an Exception Build process on the new file.

Refer to Table 8-10 for complete descriptions of all Digital point parameters.

See Figure 8-24 an Exception Build file example.

Continued on next page

8.2.2.15 Digital Output Point - 5-State Latched, continued

Table 8-23 5-State Latched Digital Output Point Parameter Entries

Parameter	Entry	Required?	Notes
NAME		Yes	See PKGNAME(4) notes below Application Specific - Refer to the User Manual section for any required packages
UNIT		Yes	
CLSLOTS	1	Yes	
NOPKG	3 or 4	Yes	
NUMSWTCH	2	Yes	
PKGNAME(1)	CLM_DMD3	Yes	
PKGNAME(2)	DIG5CDS	Yes	
PKGNAME(3)	DIG5CDS1	Yes	
PKGNAME(4)	See Notes	-	
S1NSTATE	5	Yes	Enter text for the ordinal 0 (or top) state Enter text for the ordinal 1 state Enter text for the ordinal 2 (normally stop) state Enter text for the ordinal 3 (between or moving) state Ordinal 4 (or bottom) state text. This state must be the BAD state. Key Access Level for command changes. Default is ENGINEER.
S1STATE(0)	See Notes	Yes	
S1STATE(1)	See Notes	Yes	
S1STATE(2)	See Notes	Yes	
S1STATE(3)	See Notes	Yes	
S1STATE(4)	BAD	Yes	
S1ACCLVL	Operator Supervisor Engineer Program	No	
S2NSTATE	2	Yes	Selected state for "resetting" the point. Default state. Selecting has no impact. Key Access Level for clearing switch alarms and setting commanded state = current state. This setting should match the S1ACCLVL. Default is ENGINEER.
S2STATE(0)	CLEAR	Yes	
S2STATE(1)	NOACTION	Yes	
S2ACCLVL	Operator Supervisor Engineer Program	No	
PERIOD	NOPERIOD	Yes	CLM Digital points are PPS'd for execution by data received by exception input or the CLM Point Interface Table (PIT) scheduler.

Continued on next page

8.2.2.15 Digital Output Point - 5-State Latched, continued

Table 8-23 5-State Latched Digital Output Point Parameter Requirements, continued

Parameter	Entry	Required?	Notes
D1_0	\$ord0-\$ord1	No	Not Used in the Digital Output type.
D1_1	\$ord0-\$ord1	No	Not Used in the Digital Output type.
D2D1_01	\$ord0-\$ord3	Yes	Enter the ordinal state that maps to the condition where bit 1 (D1)=ON and bit 2 (D2)=OFF
D2D1_10	\$ord0-\$ord3	Yes	Enter the ordinal state that maps to the condition where bit 1 (D1)=OFF and bit 2 (D2)=ON
D2D1_11	\$ord0-\$ord3	Yes	Enter the ordinal state that maps to the condition where bit 1 (D1)=ON and bit 2 (D2)=ON
D2D1_00	\$ord0-\$ord3	Yes	Enter the ordinal state that maps to the condition where bit 1 (D1)=OFF and bit 2 (D2)=OFF
NUMCMDST	3	Yes	Set to 3 in all cases. OPEN, CLOSED, AND STOP used in the 5-state
ALMOPT	0	No	Default = 0 for operational alarms only.
ALMUNCHG	ON or OFF	No	Default = OFF. Set to ON to alarm uncommanded state changes.
PVNORMAL	\$ord0-\$ord1	Yes IF ALMOPT=1	Not required when ALMOPT = 0. Required when ALMOPT = 1.
NODINPTS	0	Yes	Number of Digital Input signals is 0 for Digital output type
FBTIME	See Notes TT.tt	Yes	Feedback Timeouts. Maximum times (in secs) allowed for the commanded state changes, where TT=total time, tt=no movement time. Refer to Table 8-10.
TIMEOP	OFF	Yes	Set to OFF for "Latched" outputs.
NODOPTS	3	Yes	Number of Digital output signals.
ST0_OP1	ON or OFF	Yes	For ordinal state 0, bit 1. Set to ON for a logical "1", OFF for a logical "0".
ST0_OP2	ON or OFF	Yes	For ordinal state 0, bit 2. IF NODOPTS = 2, set to ON for a logical "1", OFF for a logical "0".
ST0_OP3	ON or OFF	Yes	For ordinal state 0, bit 3. IF NODOPTS = 3, set to ON for a logical "1", OFF for a logical "0".

Continued on next page

8.2.2.15 Digital Output Point - 5-State Latched, continued

Table 8-23 5-State Latched Digital Output Point Parameter Requirements, continued

Parameter	Entry	Required?	Notes
ST1_OP1	ON or OFF	Yes	For ordinal state 1, bit 1. Set to ON for a logical "1", OFF for a logical "0".
ST1_OP2	ON or OFF	Yes	For ordinal state 1, bit 2. IF NODOPTS = 2, set to ON for a logical "1", OFF for a logical "0".
ST1_OP3	ON or OFF	Yes	For ordinal state 1, bit 3. IF NODOPTS = 3, set to ON for a logical "1", OFF for a logical "0".
ST2_OP1	ON or OFF	Yes	For ordinal state 2, bit 1. Set to ON for a logical "1", OFF for a logical "0".
ST2_OP2	ON or OFF	Yes	For ordinal state 2, bit 2. IF NODOPTS = 2, set to ON for a logical "1", OFF for a logical "0".
ST2_OP3	ON or OFF	Yes	For ordinal state 2, bit 3. IF NODOPTS = 3, set to ON for a logical "1", OFF for a logical "0".
STOP_ST	0-2	Yes	Ordinal state assigned to the STOP command and position.
MOVE_ST	3	Yes	Ordinal state assigned to the MOVING or INBTWEEN position.
UNCLRTIM	-1, 0, or 1- <i>n</i>	Yes IF ALMUNCHG= ON	Configurable Clear of the Uncommanded Change alarm. IF ALMUNCHG = ON, set to: -1 for Operator clear only, 0 for automatic clear on next point execution 1- <i>n</i> for automatic clear after <i>n</i> seconds.

Continued on next page

8.2.2.15 Digital Output Point - 5-State Latched, continued

Build Example

The user wants to build a Digital point to emulate a valve with stop capability. The valve requires output signal #1 = ON for CLOSE, output signal #2 = ON for OPEN, and output signal #3 = ON for STOP. There are no limit switch feedback signals, so the outputs must be verified through the feedbacks. It takes approximately 45 seconds for the valve to completely transition from one stop to the other. If the valve has not started moving after 15 seconds, then there is a problem or a jam. The operator needs to be able to command the valve and it cannot be started or stopped at the valve (no uncommanded change). The interface is through a Modbus interface to a PLC and therefore no additional CDS packages are required. Figure 8-24 below shows an example of an Exception Build file (in LCN R500 format) that could be used to create this image point.

Figure 8-24 Build Example for 5-State Latched Digital Output

```
{SCREEN ENTITY VALVE_1( )      }
&T SWITCHAM
&N VALVE_1
UNIT = C1
PTDESC ="VALVE 1 CONTROL      "
KEYWORD ="      "
CLSLOTS = 1
NOPKG = 3
NUMSWTCH = 2                  {**NOTE: Will cause CL LINK ERROR if not set to 2**}
PRIMMOD = _____
SALMDSC1 ="      "
SALMDSC2 ="      "
SALMDSC3 ="      "
BCLEALPR = LOW                {**NOTE: R500 And Later**}
BCLFALPR = LOW                {**NOTE: R500 And Later**}
CLEALMPR = LOW                {**NOTE: R500 And Later**}
CLFALMPR = LOW                {**NOTE: R500 And Later**}
CNFERRPR = LOW                {**NOTE: R500 And Later**}
SWALM1PR = LOW                {**NOTE: R500 And Later**}
SWALM2PR = LOW                {**NOTE: R500 And Later**}
SWALM3PR = LOW                {**NOTE: R500 And Later**}
ALENBST = ENABLE
PKGNAME(1) ="CLM_DMD3"
PKGNAME(2) ="DIG5CDS "
PKGNAME(3) ="DIG5CDS1"
SINSTATE = 5
S1STATES(0) ="OPEN " {**NOTE: User-entered text**}
S1STATES(1) ="CLOSE" {**NOTE: User-entered text**}
S1STATES(2) ="STOP  "
S1STATES(3) ="MOVING "
S1STATES(4) ="BAD  "
```

Continued on next page

8.2.2.15 Digital Output Point - 5-State Latched, continued

Figure 8-24 Build Example for 5-State Latched Digital Output , continued

```
S1ACCLVL = OPERATOR
S1CURSTS = CLOSE
S1REQSTS = CLOSE
S2NSTATE = 2
S2STATES(0) ="CLEAR  "
S2STATES(1) ="NOACTION"
S2ACCLVL = OPERATOR
S2CURSTS = NOACTION
S2REQSTS = NOACTION
SIPPASN = OFF
PERIOD = NOPERIOD          {**NOTE: Scheduled and driven by associated PIT**}
CLM_PORT ="      "
CLM_DEV  ="      "
CLM_PIT  ="      "
NODINPTS = 0
D1_0   = $ORD0
D1_1   = $ORD1
D2D1_01 = $ORD1           {**CLOSE**}
D2D1_10 = $ORD0           {**OPEN**}
D2D1_00 = $ORD2           {**MOVING or STOP**}
D2D1_11 = $ORD3           {**BAD**}
NUMCMDST = 3
ALMOPT  = 0
ALMUNCHG = OFF
PVNORMAL = $ORD1
FBTIME  = 45.15
TIMEOP  = OFF             {**NOTE: MUST be OFF for Latched output signals**}
NODOPTS = 3
ST0_OP1 = OFF             {**NOTE: OPEN state sets output signal #2 to ON**}
ST0_OP2 = ON
ST0_OP3 = OFF
ST1_OP1 = ON              {**NOTE: CLOSE state sets output signal #1 to ON**}
ST1_OP2 = OFF
ST1_OP3 = OFF
ST2_OP1 = ON              {**NOTE: STOP state sets output signal #3 to ON**}
ST2_OP2 = OFF
ST2_OP3 = OFF
STOP_ST = 2
MOVE_ST = 3
UNCLRTIM = -1
UNCCLEAR = OFF
DEV_STA  = OFF
PT_FAIL  = OFF
CMDCHG   = OFF
ALMTYPE1 = A_NONE
ALMTYPE2 = A_NONE
ALMTYPE3 = A_NONE
```


8.2.2.15 Digital Output Point - 5-State Latched, continued

Figure 8-24 Build Example for 5-State Latched Digital Output , continued

```
INITLIZD = OFF
TIMER   = 0
DATA(1) = 0
DATA(2) = 0
DATA(3) = 0
DATA(4) = 0
DATA(5) = 0
DATA(6) = 0
DATA(7) = 0
DATA(8) = 0
DATA(9) = 0
DATA(10) = 0
```

Continued on next page

8.2.2.15 Digital Output Point - 5-State Latched, continued

Attach the CL Block

All CLM Digital points require a CL Block. The 5-State Digital Output uses the DIG5CL.AO block included with the CLM Platform release media. Use the EC file included to link the block to the 5-State point.

From the COMMAND PROCESSOR window of the Engineering Personality:

```
EC DIG5.EC pppppppp
      where
      pppppppp is the 5-State point name
```

NOTE: 1) Be sure that your pathnames are set for the CL SOURCE/OBJ to point to the proper directory.
2) When executing this EC file for the first time, it is normal to get an error on the UNLINK since the block has not yet been attached.

Link Caution!

If the number of switches (NUMSWTCH) on the image point is not set to 2, the following error will be issued by the Linker.

Error during attempt to get parameter information; PA Status: END OF USER SPEC

Correct and rebuild the image point before attempting to re-link.

Activate the Point

At the Detail Display of the Digital point, set the PTEXECST = ACTIVE. You may need to PROCESS the point again to clear the RESTART state to None. The S1REQSTS and S1CURSTS should agree at this point, but if not, select the S2REQSTS, select the CLEAR option and press ENTER.

Map to the Device (See Application User Manual)

The build of the image is complete from the AM side. The next step is to assign the input and/or output signals to the required parameters. This is done through the CLM Configuration Builder and is application-specific. This means that each application will map to the same image point parameters, but the format and conversion process for each specific application will probably vary. Refer to the earlier sections on the CLM Configuration Builder in this manual and the specific application user manual for instructions on mapping your particular device.

Continued on next page

8.2.2.15 Digital Output Point - 5-State Latched, continued

Command the Point

The Digital point may command an output through the S1REQSTS parameter of the image point. *THE DIGITAL POINT MUST HAVE BEEN MAPPED ALREADY THROUGH THE CLM CONFIGURATION BUILDER.*

The steps to command an output are:

- Select the S1REQSTS parameter

- Select the desired state

- Press ENTER

This will PPS the point and cause the CL to execute, sending the appropriate OPn parameter values to the device.

IMPORTANT! A mapped Digital point will not complete the S1REQSTS store if the associated device (NDT) is Out of Service or in the HARD-FAIL condition. This may be checked at the point by examining the DEV_STA parameter. If the value is OFF, then S1REQSTS stores will not complete and the operator will not see the requested change at the Detail Display.

Clear the Point

After certain alarms or other conditions, it may become necessary to clear an existing alarm(s) and/or line up the S1REQSTS and S1CURSTS parameters. This can be done through the second switch on the Digital point. The steps to perform this action are:

- Select the S2REQSTS parameter

- Select the CLEAR state

- Press ENTER

This action will have the impact of setting the S1REQSTS = S1CURSTS, clearing all existing switch alarms, and re-issuing new alarms based on current conditions.

Operation

The 5 State Digital Output point emulates a Motor-Operated-Valve (MOV) operation with no inputs. The feedback states reflect an OPEN/CLOSE condition (\$ord0-\$ord2), a STOP condition (STOP_ST), a MOVING or INBTWEEN condition (\$ord3), and a BAD or COMM_ERR condition (\$ord4). The D1 and D2 are not used since there are no separate input signals, the OP1, OP2 and OP3 parameters correspond to the pulsed output signals required to drive the Digital device, and the FB1, FB2, and FB3 parameters correspond to the feedback signals required for output verification.

When a STOP command is issued after an OPEN/CLOSE command, the current state will be set to STOP if the S1REQSTS = STOP and the input signals D1 and D2 indicate the INBTWEEN state. From this point, either OPEN or CLOSE may be commanded. Uncommanded change for the STOP command is supported by the CL. Since no indicator for the STOP is returned in most cases, an uncommanded change to MOVING/INBTWEEN will set the feedback timer. If the state has not reached a limit value by the time that the timer has elapsed, the state is set to STOP and, if configured, an uncommanded change alarm is issued.

The FBTIME timer is actually two timers that monitor two conditions. When the operator change command is generated, the first timer (fractional part of FBTIME * 100) is set. If the input state has not changed to MOVING/INBTWEEN when this timer has elapsed, a NO_MOVE alarm is issued. If it has, the second timer (the difference between the whole part of FBTIME and the value of the first timer) is initiated. If the input state has not changed to the requested state when the second timer has elapsed, then a CMDDIS (command disagree) alarm is issued.

Alarms may be cleared or the requested state set equal to the current state by selecting the S2REQSTS parameter from the Detail Display for the point and selecting the CLEAR option.

Application

Digital device type examples:

- Pump motor monitor and control
- Valve control
- Devices requiring 0 inputs and 3 pulsed output signals

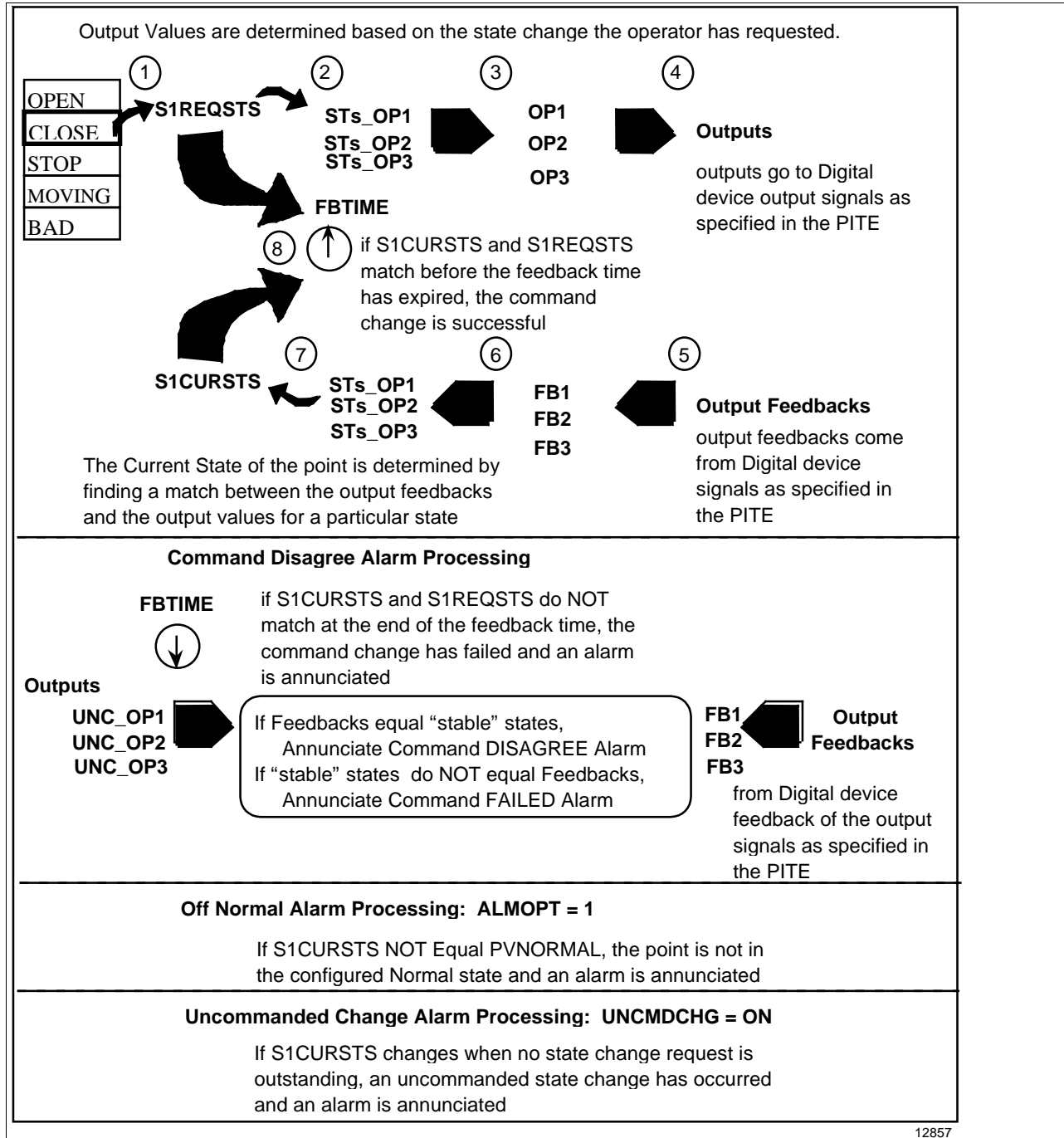
Continued on next page

8.2.2.16 Digital Output Point, 5-State Pulsed, continued

Operational Flow

Following is a depiction of how a 5-State Digital Output operates. It shows a Digital Output with three output signals that are **Pulsed** Outputs and no input signals.

Figure 8-25 Operational Flow for 5-State Pulsed Digital Output



Continued on next page

8.2.2.16 Digital Output Point - 5-State Pulsed, continued

I/O Configuration Options

Number of Inputs	Number of Outputs	Max Cmd States
0	3	3

Alarming

The following is a list of alarms that may be issued by this Digital point type. Refer to Table 8-9 for specific alarm definition.

- OUT_SERV
- PT_FAIL
- CFG_ERR_*n*
- INVCMD
- CMDFAIL
- CMDDIS
- NO_MOVE
- UNC_CHG (If configured through ALMUNCHG)
- BAD_PV
- BADSTATE

Build the Point

The 5-State Digital point is based on the AM Switch point. There are two methods of building this point type. Both of these methods use the Data Entity Builder (DEB) and the data entry requirements listed in Table 8-24 below.

- 1) Select the AM SWITCH point type and enter data as listed below.
or
- 2) Copy the DOUT5P.EB Exception Build file (included and installed with the Platform release media) to another file for editing.

Edit the new Exception Build file using the data listed below as input for each of the points to be included within the file. Cut and Paste as required.

Perform an Exception Build process on the new file.

Refer to Table 8-10 for complete descriptions of all Digital point parameters.

See Figure 8-26 an Exception Build file example.

Continued on next page

8.2.2.16 Digital Output Point - 5-State Pulsed, continued

Table 8-24 5-State Pulsed Digital Output Point Parameter Entries

Parameter	Entry	Required?	Notes
NAME		Yes	See PKGNAME(4) notes below
UNIT		Yes	
CLSLOTS	1	Yes	
NOPKG	3 or 4	Yes	
NUMSWTCH	2	Yes	
PKGNAME(1)	CLM_DMD3	Yes	
PKGNAME(2)	DIG5CDS	Yes	
PKGNAME(3)	DIG5CDS1	Yes	
PKGNAME(4)	See Notes	-	
S1NSTATE	5	Yes	Application Specific - Refer to the User Manual section for any required packages
S1STATE(0)	See Notes	Yes	
S1STATE(1)	See Notes	Yes	
S1STATE(2)	See Notes	Yes	
S1STATE(3)	See Notes	Yes	
S1STATE(4)	BAD	Yes	
S1ACCLVL	Operator Supervisor Engineer Program	No	
S2NSTATE	2	Yes	Enter text for the ordinal 0 (or top) state
S2STATE(0)	CLEAR	Yes	
S2STATE(1)	NOACTION	Yes	
S2ACCLVL	Operator Supervisor Engineer Program	No	
PERIOD	NOPERIOD	Yes	Enter text for the ordinal 2 (normally stop) state
			Enter text for the ordinal 3 (between or moving) state
			Ordinal 4 (or bottom) state text. This state must be the BAD state.
			Key Access Level for command changes. Default is ENGINEER.
			Selected state for "resetting" the point.
			Default state. Selecting has no impact.
			Key Access Level for clearing switch alarms and setting commanded state = current state. This setting should match the S1ACCLVL. Default is ENGINEER.
			CLM Digital points are PPS'd for execution by data received by exception input or the CLM Point Interface Table (PIT) scheduler.

Continued on next page

8.2.2.16 Digital Output Point - 5-State Pulsed, continued

Table 8-24 5-State Pulsed Digital Output Point Parameter Requirements, continued

Parameter	Entry	Required?	Notes
D1_0	\$ord0-\$ord1	No	Not Used in the Digital Output type.
D1_1	\$ord0-\$ord1	No	Not Used in the Digital Output type.
D2D1_01	\$ord0-\$ord3	Yes	Enter the ordinal state that maps to the condition where bit 1 (D1)=ON and bit 2 (D2)=OFF
D2D1_10	\$ord0-\$ord3	Yes	Enter the ordinal state that maps to the condition where bit 1 (D1)=OFF and bit 2 (D2)=ON
D2D1_11	\$ord0-\$ord3	Yes	Enter the ordinal state that maps to the condition where bit 1 (D1)=ON and bit 2 (D2)=ON
D2D1_00	\$ord0-\$ord3	Yes	Enter the ordinal state that maps to the condition where bit 1 (D1)=OFF and bit 2 (D2)=OFF
NUMCMDST	3	Yes	Set to 3 in all cases. OPEN, CLOSED, AND STOP used in the 5-state
ALMOPT	0	No	Default = 0 for operational alarms only.
ALMUNCHG	ON or OFF	No	Default = OFF. Set to ON to alarm uncommanded state changes.
PVNORMAL	\$ord0-\$ord1	Yes IF ALMOPT=1	Not required when ALMOPT = 0. Required when ALMOPT = 1.
NODINPTS	0	Yes	Number of Digital Input signals is 0 for Digital output type
FBTIME	See Notes TT.tt	Yes	Feedback Timeouts. Maximum times (in secs) allowed for the commanded state changes, where TT=total time, tt=no movement time. Refer to Table 8-10.
TIMEOP	ON	Yes	Set to ON for "Pulsed" outputs.
NODOPTS	3	Yes	Number of Digital output signals.
ST0_OP1	ON or OFF	Yes	For ordinal state 0, bit 1. Set to ON for a logical "1", OFF for a logical "0".
ST0_OP2	ON or OFF	Yes	For ordinal state 0, bit 2. IF NODOPTS = 2, set to ON for a logical "1", OFF for a logical "0".
ST0_OP3	ON or OFF	Yes	For ordinal state 0, bit 3. IF NODOPTS = 3, set to ON for a logical "1", OFF for a logical "0".

Continued on next page

8.2.2.16 Digital Output Point - 5-State Pulsed, continued

Table 8-24 5-State Pulsed Digital Output Point Parameter Requirements, continued

Parameter	Entry	Required?	Notes
ST1_OP1	ON or OFF	Yes	For ordinal state 1, bit 1. Set to ON for a logical "1", OFF for a logical "0".
ST1_OP2	ON or OFF	Yes	For ordinal state 1, bit 2. IF NODOPTS = 2, set to ON for a logical "1", OFF for a logical "0".
ST1_OP3	ON or OFF	Yes	For ordinal state 1, bit 3. IF NODOPTS = 3, set to ON for a logical "1", OFF for a logical "0".
ST2_OP1	ON or OFF	Yes	For ordinal state 2, bit 1. Set to ON for a logical "1", OFF for a logical "0".
ST2_OP2	ON or OFF	Yes	For ordinal state 2, bit 2. IF NODOPTS = 2, set to ON for a logical "1", OFF for a logical "0".
ST2_OP3	ON or OFF	Yes	For ordinal state 2, bit 3. IF NODOPTS = 3, set to ON for a logical "1", OFF for a logical "0".
STOP_ST	0-2	Yes	Ordinal state assigned to the STOP command and position.
MOVE_ST	3	Yes	Ordinal state assigned to the MOVING or INBTWEEN position.
UNC_OP1	ON or OFF	Yes	For "stable" state ,bit 1. Set to ON for a logical "1", OFF for a logical "0".
UNC_OP2	ON or OFF	Yes	For "stable" state, bit 2. Set to ON for a logical "1", OFF for a logical "0".
UNC_OP3	ON or OFF	Yes	For "stable" state, bit 3. Set to ON for a logical "1", OFF for a logical "0".
UNCLRTIM	-1, 0, or 1- <i>n</i>	Yes IF ALMUNCHG= ON	Configurable Clear of the Uncommanded Change alarm. IF ALMUNCHG = ON, set to: -1 for Operator clear only, 0 for automatic clear on next point execution 1- <i>n</i> for automatic clear after <i>n</i> seconds.

Continued on next page

8.2.2.16 Digital Output Point - 5-State Pulsed, continued

Build Example

The user wants to build a Digital point to emulate a valve with stop capability. The valve requires output signal #1 = ON for CLOSE, output signal #2 = ON for OPEN, and output signal #3 = ON for STOP. The digital device driver board pulses each output signal to ON for 2 seconds and then returns to a stable state of OFF. There are no limit switch feedback signals, so the outputs must be verified through the feedbacks. It takes approximately 45 seconds for the valve to completely transition from one stop to the other. If the valve has not started moving after 15 seconds, then there is a problem or a jam. The operator needs to be able to command the valve and it cannot be started or stopped at the valve (no uncommanded change). The interface is through a Modbus interface to a PLC and therefore no additional CDS packages are required. Figure 8-26 below shows an example of an Exception Build file (in LCN R500 format) that could be used to create this image point.

Figure 8-26 Build Example for 5-State Pulsed Digital Output

```
{SCREEN ENTITY VALVE_1( ) }
&T SWITCHAM
&N VALVE_1
UNIT = C1
PTDESC ="VALVE 1 CONTROL  "
KEYWORD ="  "
CLSLOTS = 1
NOPKG = 3
NUMSWTCH = 2          {**NOTE: Will cause CL LINK ERROR if not set to 2**}
PRIMMOD = _____
SALMDSC1 ="  "
SALMDSC2 ="  "
SALMDSC3 ="  "
BCLEALPR = LOW        {**NOTE: R500 And Later**}
BCLFALPR = LOW        {**NOTE: R500 And Later**}
CLEALMPR = LOW        {**NOTE: R500 And Later**}
CLFALMPR = LOW        {**NOTE: R500 And Later**}
CNFERRPR = LOW        {**NOTE: R500 And Later**}
SWALM1PR = LOW        {**NOTE: R500 And Later**}
SWALM2PR = LOW        {**NOTE: R500 And Later**}
SWALM3PR = LOW        {**NOTE: R500 And Later**}
ALENBST = ENABLE
PKGNAME(1) ="CLM_DMD3"
PKGNAME(2) ="DIG5CDS "
PKGNAME(3) ="DIG5CDS1"
SINSTATE = 5
```

Continued on next page

8.2.2.16 Digital Output Point - 5-State Pulsed, continued

Figure 8-26 Build Example for 5-State Pulsed Digital Output , continued

```
S1STATES(0) = "OPEN " {**NOTE: User-entered text**}
S1STATES(1) = "CLOSE" {**NOTE: User-entered text**}
S1STATES(2) = "STOP "
S1STATES(3) = "MOVING "
S1STATES(4) = "BAD "
S1ACCLVL = OPERATOR
S1CURSTS = CLOSE
S1REQSTS = CLOSE
S2NSTATE = 2
S2STATES(0) = "CLEAR "
S2STATES(1) = "NOACTION"
S2ACCLVL = OPERATOR
S2CURSTS = NOACTION
S2REQSTS = NOACTION
$IPPASN = OFF
PERIOD = NOPERIOD {**NOTE: Scheduled and driven by associated PIT**}
CLM_PORT = " "
CLM_DEV = " "
CLM_PIT = " "
NODINPTS = 0
D1_0 = $ORD0
D1_1 = $ORD1
D2D1_01 = $ORD1 {**CLOSE**}
D2D1_10 = $ORD0 {**OPEN**}
D2D1_00 = $ORD2 {**MOVING or STOP**}
D2D1_11 = $ORD3 {**BAD**}
NUMCMDST = 3
ALMOPT = 0
ALMUNCHG = OFF
PVNORMAL = $ORD1
FBTIME = 45.15
TIMEOP = ON {**NOTE: MUST be ON for pulsed output signals**}
NODOPTS = 3
ST0_OP1 = OFF {**NOTE: OPEN state sets output signal #2 to ON**}
ST0_OP2 = ON
ST0_OP3 = OFF
ST1_OP1 = ON {**NOTE: CLOSE state sets output signal #1 to ON**}
ST1_OP2 = OFF
ST1_OP3 = OFF
ST2_OP1 = ON {**NOTE: STOP state sets output signal #3 to ON**}
ST2_OP2 = OFF
ST2_OP3 = OFF
STOP_ST = 2
MOVE_ST = 3
UNCLRTIM = -1
UNCCLEAR = OFF
```

Continued on next page

8.2.2.16 Digital Output Point - 5-State Pulsed, continued

Figure 8-26 Build Example for 5-State Pulsed Digital Output , continued

```
UNC_OP1 = ON
UNC_OP2 = OFF
UNC_OP3 = OFF
DEV_STA = OFF
PT_FAIL = OFF
CMDCHG = OFF
ALMTYPE1 = A_NONE
ALMTYPE2 = A_NONE
ALMTYPE3 = A_NONE
INITLIZD = OFF
TIMER = 0
DATA(1) = 0
DATA(2) = 0
DATA(3) = 0
DATA(4) = 0
DATA(5) = 0
DATA(6) = 0
DATA(7) = 0
DATA(8) = 0
DATA(9) = 0
DATA(10) = 0
```

Continued on next page

8.2.2.16 Digital Output Point - 5-State Pulsed, continued

Attach the CL Block

All CLM Digital points require a CL Block. The 5-State Digital Output uses the DIG5CL.AO block included with the CLM Platform release media. Use the EC file included to link the block to the 5-State point.

From the COMMAND PROCESSOR window of the Engineering Personality:

EC DIG5.EC *pppppppp*

where

pppppppp is the 5-State point name

NOTE: 1) Be sure that your pathnames are set for the CL SOURCE/OBJ to point to the proper directory.
2) When executing this EC file for the first time, it is normal to get an error on the UNLINK since the block has not yet been attached.

Link Caution!

If the number of switches (NUMSWTCH) on the image point is not set to 2, the following error will be issued by the Linker.

Error during attempt to get parameter information; PA Status: END OF USER SPEC

Correct and rebuild the image point before attempting to re-link.

Activate the Point

At the Detail Display of the Digital point, set the PTEXECST = ACTIVE. You may need to PROCESS the point again to clear the RESTART state to None. The S1REQSTS and S1CURSTS should agree at this point, but if not, select the S2REQSTS, select the CLEAR option and press ENTER.

Map to the Device (See Application User Manual)

The build of the image is complete from the AM side. The next step is to assign the input and/or output signals to the required parameters. This is done through the CLM Configuration Builder and is application-specific. This means that each application will map to the same image point parameters, but the format and conversion process for each specific application will probably vary. Refer to the earlier sections on the CLM Configuration Builder in this manual and the specific application user manual for instructions on mapping your particular device.

Continued on next page

8.2.2.16 Digital Output Point - 5-State Pulsed, continued

Command the Point

The Digital point may command an output through the S1REQSTS parameter of the image point. *THE DIGITAL POINT MUST HAVE BEEN MAPPED ALREADY THROUGH THE CLM CONFIGURATION BUILDER.*

The steps to command an output are:

- Select the S1REQSTS parameter

- Select the desired state

- Press ENTER

This will PPS the point and cause the CL to execute, sending the appropriate OPn parameter values to the device.

<p>IMPORTANT! A mapped Digital point will not complete the S1REQSTS store if the associated device (NDT) is Out of Service or in the HARD-FAIL condition. This may be checked at the point by examining the DEV_STA parameter. If the value is OFF, then S1REQSTS stores will not complete and the operator will not see the requested change at the Detail Display.</p>

Clear the Point

After certain alarms or other conditions, it may become necessary to clear an existing alarm(s) and/or line up the S1REQSTS and S1CURSTS parameters. This can be done through the second switch on the Digital point. The steps to perform this action are:

- Select the S2REQSTS parameter

- Select the CLEAR state

- Press ENTER

This action will have the impact of setting the S1REQSTS = S1CURSTS, clearing all existing switch alarms, and re-issuing new alarms based on current conditions.

8.2.3 Flag Image Point Type

Flag point configuration

The AM flag point type supports a single bit input and/or output. CDS is not supported for this point type.

Flag Point Configuration

Description	Configuration
Point Type	AM Flag
PITE PV Input to:	PV

Note: The point data mapping is defined specifically within the application. This point type will support input and output of discrete data.

8.2.4 Numeric Image Point Type

Numeric point configuration

The AM numeric point type supports an analog value. CDS is not supported for the AM Numeric Point.

Numeric Point Configuration

Description	Configuration
Point Type	AM Flag
PITE PV Input to:	PV

Note: The point mapping is defined specifically within the application. This point type will support input and output of analog data.

8.2.5 Pulse Input Image Point Type

Pulse input points

A pulse input point with CDS and CL can be implemented similar to the one in the Process Manager (PM). Pulse input points always provide two related functions simultaneously: totalizing and frequency counting. Totalizing means the non-Honeywell device maintains a 32-bit accumulation (AV) for each pulse input. This counter continually rolls (i.e., there is no start, stop, or reset). Frequency counting is expressed by the parameters PVRAW and PVCALC.

$$\text{PVRAW} = \text{Delta} / \text{Elapsed Time in Seconds} = \text{Pulses Per Second}$$

PVCALC is PVRAW after characterization (e.g., barrels per hour)

$$\text{PVCALC} = \text{C1/C2} * \text{TIMEBASE} * \text{PVRAW}$$

C1 is an engineering unit conversion factor, such as barrels per gallon;
C2 is the meter factor in engineering units per pulse (e.g., gallons per pulse).

TIMEBASE is a time scale factor. The choices are seconds, minutes, or hours.

This point can serve as input to an AM regulatory point implemented with the totalizer algorithm or to an AM counter point which was first made for the 32-bit counter in the HLPIU.

The same functionality is available in selected remote devices. In those cases, it is only necessary to image the two remote signals, etc.

8.3 Custom CLM Point Types

Custom points

The CLM regulatory, switch, and custom points can be extended by the addition of CDS and CL processing blocks.

Additional parameters unique to a non-Honeywell device are placed in a CDS. Special point processing, not handled by the protocol layer, can be implemented with an appropriate CL processing block.

Typical non-standard CLM image point types that might be created are:

- ⇒ Tank data points
 - ⇒ Analyzer Stream and Component points
 - ⇒ Manual Selector Switch points,
 - ⇒ Pressure transmitter points
 - ⇒ Mass Spectrometer Data points
-

8.4 CL Interfaces

Supported CL interfaces

Three types of CL interfaces are supported by the CLM Platform. The three following sections cover each of these CL interfaces in detail.

Three Supported CL Interfaces

Subsection	Action
8.4.1	CL Interface to Non-Honeywell Device
8.4.2	CL Interface to NDT and PIT Structures
8.4.3	CL Interface to Image Point Demand Processing

Application requirements

The CLM platform alone may not provide all functions required to support a particular non-Honeywell device interface or the functions necessary to support upper level applications requirements.

Some of these requirements can be satisfied using standard CLM/AM functions, including CL programs.

Custom displays can also be used to fulfill some application requirements.

To assist in meeting special requirements, the CLM provides the CLM.SF Set Definition File, which contains the CL interfaces to low level CLM functions covered in the following three subsections.

These interface calls may be included in the CI source code by adding the following statement at the beginning of the code.

```
%INCLUDE_SET CLM;
```

8.4.1 CL Interface to Non-Honeywell Device

Overview

This interface permits AM/CL programs to send and receive messages to CLM Interface board resident application tasks. This mechanism can be used to initiate special pre-programmed functions on the interface board or simply retrieve information about the Network Memory (mapping) data base. Refer to the communication and protocol specification for the object network for supported functions and their message structures.

A common function supported by most CLM communication and protocols is the blind, or transparent, communications request. These requests are passed through the communication software to the network or device and the reply returned to the CL program. The process overview is:

To send the message,

1. The CL task first calls a `reserve_buffer` routine to be used as a communication buffer.
2. The CL calls `put_field` routines as many times as it takes to pack the message.
3. Once the message is packed, the CL calls the `send_message` routine with parameters for the message.

To receive the message,

- 4) The CL first reserves another buffer or uses the previously reserved buffer as the receive buffer.
- 5) The CL fetches the reply by invoking the `receive_message` routine which returns the reply message into the receive buffer.
- 6) The CL task then calls the `get_field` routine as many times as needed to unpack the message.
- 7) The CL calls the `release_buffer` routine to release the reserved buffer to the AM memory at the end of the interface.

There are six CL calls associated with non-Honeywell device interfaces:

- `CALL CLM$RES_BUFFER`
- `CALL CLM$REL_BUFFER`
- `CALL CLM$PUT_FIELD`
- `CALL CLM$GET_FIELD`
- `CALL CLM$SEND_CL_MESSAGE`
- `CALL CLM$RECV_CL_MESSAGE`

Continued on next page

8.4.1

CL Interface to Non-Honeywell Device, Continued

CLM\$RES_BUFFER

```
CALL CLM$RES_BUFFER ( BUFFER_SIZE : IN NUMBER;  
                      RETSTS       : OUT NUMBER;  
                      BUFFER_NUM   : IN OUT NUMBER )
```

Purpose

This function is used by the CL to reserve a buffer for communication.

At the end of data transaction, a call to CLM\$REL_BUFFER must be made to free up memory that was reserved by the reserve call.

Parameters

Parameters

Explanation

BUFFER_SIZE	Size of the buffer for send or receive buffer (in bytes)
RETSTS	The return status of the reserve buffer function
BUFFER_NUM	The buffer number returned by the call; initialize to zero to pass to the call. Value is indeterminate if call unsuccessful.

Return Status Value

Value	Description
0.0	no error - call was successful; buffer reserved.
1.0	get memory error - AM does not currently have enough memory to reserve the buffer.
112.0	CL message buffer is not available - All available buffers are currently in use; call CLM\$REL_BUFFER to free up a buffer and retry.
113.0	duplicate pointer name - internal application error
114.0	global memory table full - internal application error

Continued on next page

8.4.1

CL Interface to Non-Honeywell Device, Continued

CLM\$REL_BUFFER `CALL CLM$REL_BUFFER (BUFFER_SIZE : IN NUMBER;
 RETSTS : OUT NUMBER;
 BUFFER_NUM : IN OUT NUMBER)`

Purpose

This function is used by the CL to release a buffer to AM memory at the end of the interface.

This call must have been preceded by a CLM\$RES_BUFFER call.

Parameters

Parameters	Explanation
BUFFER_SIZE	Size of the buffer to be released (in bytes)
RETSTS	The return status of the reserve buffer function
BUFFER_NUM	The buffer number (from the CLM\$RES_BUFFER call) that needs to be released.

Return Status Value

Value	Description
0.0	no error - call was successful; memory has been returned.
10.0	put memory error - internal application error
111.0	invalid buffer size - buffer size passed is not valid; probably an uninitialized or negative value.
115.0	empty buffer error - specified buffer is not reserved or has been already returned
116.0	global pointer not found - internal application error

Continued on next page

8.4.1

CL Interface to Non-Honeywell Device, Continued

CLM\$PUT_FIELD

```
CALL CLM$PUT_FIELD ( RETSTS      : OUT NUMBER;  
                     DA_STATUS   : OUT NUMBER;  
                     CHAR_POINTER : IN OUT NUMBER;  
                     BUFFER_NUM  : IN OUT NUMBER;  
                     VALUE_REF   : IN REFERENCE_N;  
                     FORMAT      : IN STRING;  
                     APPEND_STRING : IN STRING)
```

Purpose

This function takes an input value in internal form and places it at the char pointer position in the buffer. The char_pointer is incremented to the next available position in the buffer.

Parameters

Parameter	Explanation
RETSTS	The return status of the put field function
DA_STATUS	Data Access return status code from the read of the parameter containing the field data value. <i>See section 5.15.</i>
CHAR_POINTER	Current relative character (byte) offset in the buffer.
BUFFER_NUM	The number of the buffer being referenced.
VALUE_REF	Actual value to be converted and placed in the buffer. This value cannot be a self-defining enumeration.
FORMAT	The specified format of the field: byte "B" 1 byte integer "I" 2 bytes IEEE "E" 4 bytes number "R-ZZZZ9.99" logical "TEXTL1:5" string "TEXTL1:10" time "DATEMM-DD-YYENDATE" enumeration "TEXTL1:10" entity id "TEXTL1:10"
APPEND_STRING	Appended string to be added to the specified field. This field is optional. Set to null string if not used. Appended strings are added after the converted value.

Continued on next page

8.4.1

CL Interface to Non-Honeywell Device, Continued

CLM\$PUT_FIELD
Parameters,
continued

Return Status Value

Value	Description
0.0	no error - call was successful; memory has been returned.
10.0	put memory error - internal application error
111.0	invalid buffer size - buffer size passed is not valid; probably an uninitialized or negative value.
112.0	CL message buffer is not available - All available buffers are currently in use; call CLM\$REL_BUFFER to free up a buffer and retry.
113.0	duplicate pointer name - internal application error
115.0	empty buffer error - specified buffer is not reserved or has been already returned
116.0	global pointer not found - internal application error
117.0	invalid buffer number - the buffer number passed is either too large (>100) or negative.
119.0	global pointer error - internal application error
120.0	mismatch buffer size error - The amount of data already in the buffer has exceeded the buffer size.
241.0	buffer overflow - The amount of data to be put into the buffer has exceeded the buffer size.

Continued on next page

8.4.1

CL Interface to Non-Honeywell Device, Continued

CLM\$GET_FIELD

```
CALL CLM$GET_FIELD (  RETSTS      : OUT NUMBER;  
                      DA_STATUS   : OUT NUMBER;  
                      CHAR_POINTER : IN OUT NUMBER;  
                      BUFFER_NUM  : IN OUT NUMBER;  
                      VALUE_REF   : IN REFERENCE_N;  
                      FORMAT      : IN STRING;  
                      STRING_WIDTH : IN NUMBER)
```

Purpose

This function extracts an external value from a field in the buffer, optionally converts it to internal form and then stores it into VALUE_REF (output value). Characters must point to the beginning of the field. Anticipated field delimiters are: Space, Comma, Left parenthesis, Right parenthesis, Colon, Semicolon, and Carriage Return. The char_pointer is incremented to the next available position.

Parameters

Parameter	Explanation																											
RETSTS	The return status of the put field function																											
DA_STATUS	Data access error status																											
CHAR_POINTER	Character offset from the beginning of buffer																											
BUFFER_NUM	Buffer ID number																											
VALUE_REF	Output value to be converted (This value cannot be a self-defining enumeration.)																											
FORMAT	The specified format of the field: <table><tr><td>byte</td><td>“B”</td><td>1 byte</td></tr><tr><td>integer</td><td>“I”</td><td>2 byte</td></tr><tr><td>IEEE</td><td>“E”</td><td>4 byte</td></tr><tr><td>number</td><td>“R-<i>ZZZZ</i>9.99”</td><td></td></tr><tr><td>logical</td><td>“TEXTL1:5”</td><td></td></tr><tr><td>string</td><td>“TEXTL1:10”</td><td></td></tr><tr><td>time</td><td>“DATEMM-DD-YYENDATE”</td><td></td></tr><tr><td>enumeration</td><td>“TEXTL1:10”</td><td></td></tr><tr><td>entity (point ID)</td><td>“TEXTL1:10”</td><td></td></tr></table>	byte	“B”	1 byte	integer	“I”	2 byte	IEEE	“E”	4 byte	number	“R- <i>ZZZZ</i> 9.99”		logical	“TEXTL1:5”		string	“TEXTL1:10”		time	“DATEMM-DD-YYENDATE”		enumeration	“TEXTL1:10”		entity (point ID)	“TEXTL1:10”	
byte	“B”	1 byte																										
integer	“I”	2 byte																										
IEEE	“E”	4 byte																										
number	“R- <i>ZZZZ</i> 9.99”																											
logical	“TEXTL1:5”																											
string	“TEXTL1:10”																											
time	“DATEMM-DD-YYENDATE”																											
enumeration	“TEXTL1:10”																											
entity (point ID)	“TEXTL1:10”																											
STRING_WIDTH	Number of characters in string, if format is of type string.																											

Continued on next page

8.4.1

CL Interface to Non-Honeywell Device, Continued

CLM\$GET_FIELD Parameters, continued

Return Status Value

Value	Description
0.0	no error - call was successful; memory has been returned.
10.0	put memory error - internal application error
111.0	invalid buffer size - buffer size passed is not valid; probably an uninitialized or negative value.
112.0	CL message buffer is not available - All available buffers are currently in use; call CLM\$REL_BUFFER to free up a buffer and retry.
113.0	duplicate pointer name - internal application error
115.0	empty buffer error - specified buffer is not reserved or has been already returned
116.0	global pointer not found - internal application error
117.0	invalid buffer number - the buffer number passed is either too large (>100) or negative.
119.0	global pointer error - internal application error
120.0	mismatch buffer size error - The amount of data already in the buffer has exceeded the buffer size.
241.0	buffer overflow - The amount of data to be put into the buffer has exceeded the buffer size.

Note:

The following applies only to output values, not to type "string:"

Get field scans the current buffer from left to right, skipping any leading blank characters, and extracts characters until it finds a delimiter. After the extracted external value is converted to the specified internal form and stored in output value, the character pointer is adjusted to point to the character after the delimiter.

For output values of data type "string" the following special rules apply:

String width specifies the number of consecutive characters to be moved into output value. Any field delimiters found among these characters are considered to be part of the string. Character pointer is returned pointing at the character immediately following the last character transferred. Note that, depending on the buffer structure, the character pointer could wind up pointing to data, or to a field delimiter.

Buffer number must be the value returned on a previous CLM\$RES_BUFFER call.

Continued on next page

8.4.1

CL Interface to Non-Honeywell Device, Continued

```
CLM$SEND_CL_MESSAGE  CALL CLM$SEND_CL_MESSAGE (  PORT          : IN NUMBER ;
                                                    BUFFER_NUM   : IN OUT NUMBER ;
                                                    MSG_LEN      : IN NUMBER ;
                                                    MSG_ID       : IN NUMBER ;
                                                    RETSTS       : OUT NUMBER )
```

Purpose

This function is used by the CL to send a message to the CLMI board identified by the port code and to obtain the message identifier that will be used in the receive message call to identify the proper response message.

Parameters

Parameter	Explanation
RETSTS	The return status of the send_cl_message function.
PORT	Port number of the CLMI task that should receive the message. 1-16 = CLMI #1 17-32 = CLMI #2
BUFFER_NUM	Buffer id which contains the message to be sent.
MSG_LEN	Length, in bytes , of the message to be sent.
MSG_ID	The newly generated MSG_ID returned to the CL caller to be used in the subsequent receive message call.

Return Status Value

Value	Description
0.0	no error - call was successful; memory has been returned.
10.0	put memory error - internal application error
111.0	invalid buffer size - buffer size passed is not valid; probably an uninitialized or negative value.
112.0	CL message buffer is not available - All available buffers are currently in use; call CLM\$REL_BUFFER to free up a buffer and retry.
113.0	duplicate pointer name - internal application error
115.0	empty buffer error - specified buffer is not reserved or has been already returned
116.0	global pointer not found - internal application error
117.0	invalid buffer number - the buffer number passed is either too large (>100) or negative.
119.0	global pointer error - internal application error
120.0	mismatch buffer size error - The amount of data already in the buffer has exceeded the buffer size.
241.0	buffer overflow - The amount of data to be put into the buffer has exceeded the buffer size.

Continued on next page

8.4.1

CL Interface to Non-Honeywell Device, Continued

```
CLM$RECV_CL_MESSAGE      CALL CLM$RECV_CL_MESSAGE ( MSG_ID      : IN NUMBER;  
                                                             BUFFER_NUM : IN OUT NUMBER;  
                                                             MSG_LEN   : IN NUMBER;  
                                                             WAIT_CYCLES : IN NUMBER;  
                                                             RETSTS     : OUT NUMBER )
```

Purpose

This function obtains the status for a pending CL request to the CLMI board. After the send_cl_message call has been made, the CL block checks the request status periodically (based on the period of the image point) by calling recv_cl_message. A status returned of 248 indicates that the message has not yet been processed by the CLMI task and must be checked again until completion is determined. A return status of 0 indicates the successful completion of the send process and the reply is given and stored in the indicated buffer. All other return status indications are for unsuccessful responses.

Parameters

Parameter	Explanation
RETSTS	The return status of the send message function.
MSG_ID	The message identifier for the message to be checked. This MSG_ID was received from the previously executed send message routine. This value must be retained until the receive message is satisfied.
BUFFER_NUM	Buffer ID which contains the message to be sent.
MSG_LEN	Length, in bytes , of the message received.
WAIT_CYCLES	Wait time in seconds. This is the time that the recv_cl_message will wait before returning a status back to the calling CL block.

Continued on next page

8.4.1

CL Interface to Non-Honeywell Device, Continued

CLM\$RECV_CL_MESSAGE

Parameters.

continued

Return Status Value

Value	Description
0.0	no error - call was successful; memory has been returned.
10.0	put memory error - internal application error
111.0	invalid buffer size - buffer size passed is not valid; probably an uninitialized or negative value.
112.0	CL message buffer is not available - All available buffers are currently in use; call CLM\$REL_BUFFER to free up a buffer and retry.
113.0	duplicate pointer name - internal application error
115.0	empty buffer error - specified buffer is not reserved or has been already returned
116.0	global pointer not found - internal application error
117.0	invalid buffer number - the buffer number passed is either too large (>100) or negative.
119.0	global pointer error - internal application error
120.0	mismatch buffer size error - The amount of data already in the buffer has exceeded the buffer size.
241.0	buffer overflow - The amount of data to be put into the buffer has exceeded the buffer size.
248.0	message timeout - The call has waited the specified length of time and the CLMI board/application has not yet responded.

8.4.2 CL Interface to NDT and PIT Structures

Overview

This interface provides a CL program interface to the NDT and PIT structures. The interface permits CL blocks to read and modify selected data in these entries. The following functions are provided and can be called by the user-developed CL modules:

There are four CL task calls associated with a non-Honeywell device interface:

- CALL CLM\$Get_NDT_Parameter
 - CALL CLM\$Put_NDT_Parameter
 - CALL CLM\$Get_PIT_Parameter
 - CALL CLM\$Put_PIT_Parameter
-

8.4.2

CL Interface to NDT and PIT Structures, Continued

CLM\$Get_NDT_Parameter

```
CALL CLM$Get_NDT_Parameter ( NDT_NAME  : IN STRING ;
                             PARM_ID    : IN NUMBER ;
                             PARM_VAL   : OUT NUMBER ;
                             RET_STAT   : OUT NUMBER
                             ) ; BACKGROUND ;
```

Parameters

CLM\$Get_NDT_Parameter Parameters	
Parameter	Explanation
NDT_NAME	The ASCII name of the NDT entry whose values are of interest.
PARM_ID	The index of the parameter ID from the table below that corresponds to the parameter of interest.
PARM_VAL	The returned value of the parameter if the NDT name and parameter ID are both valid.
RET_STAT	The returned error status for the routine. If any problem was encountered attempting to get the value, a non-zero value is returned.

Purpose

To obtain the value of the given NDT parameter. The NDT is specified by the ASCII name of the NDT entry. The parameter is specified by the index of the corresponding parameter ID. The parameter ID is defined as follows:

<u>INDEX</u>	<u>NDT PARAMETER ID</u>
01-30	DET_STS(1-30)
31-38	ADDRESS(1-8)
39	UP_STS
40	DWN_STS
41	STS (GOOD=0; ERROR=1; FAILED=2; ACK=3; UNKNOWN=4; PRIMARY UNAVAILABLE=5
42	STA (IN_SERVICE=0; SYNC=1; FAILED=2; OUT_OF_SERVICE=3

If the NDT name is not found or if the parameter index is invalid, the value returned by the procedure is undefined.

Continued on next page

8.4.2

CL Interface to NDT and PIT Structures, Continued

CLM\$Put_NDT_ Parameter

```
CALL CLM$Put_NDT_Parameter ( NDT_NAME   : IN STRING;  
                             PARM_ID    : IN NUMBER;  
                             ARM_VAL    : OUT NUMBER;  
                             RET_STAT   : OUT NUMBER  
                             ) ;BACKGROUND;
```

Parameters

CLM\$Put_NDT_Parameter Parameters

Parameter	Explanation
NDT_NAME	The ASCII name of the NDT entry whose values are to be changed.
PARM_ID	The index of the parameter ID from the table below that corresponds to the parameter to be changed.
PARM_VAL	The value to be placed into the parameter if the NDT name and parameter ID are both valid.
RET_STAT	The returned error status for the routine. If any problem is encountered attempting to store the value, a non-zero value is returned.

Purpose

To change the value of the given NDT parameter. The NDT is specified by the ASCII name of the NDT entry. The parameter is specified by the index of the corresponding parameter ID. The parameter ID is defined as follows:

<u>INDEX</u>	<u>NDT PARAMETER ID</u>
06-30	DET_STS(6-30)

Note: DET_STS(1-5) are reserved and used by the CLM Platform and are not available for modification.

If the NDT name is not found or if the parameter is invalid, the value returned by the procedure is undefined.

Continued on next page

8.4.2

CL Interface to NDT and PIT Structures, Continued

CLM\$Get_PIT_ Parameter

```
CALL CLM$Get_PIT_Parameter ( NDT_NAME  : IN STRING;  
                             PARM_ID    : IN NUMBER;  
                             ARM_VAL    : OUT NUMBER;  
                             RET_STAT   : OUT NUMBER  
                             ) ; BACKGROUND ;
```

Parameters

CLM\$Get_PIT_Parameter Parameters

Parameter	Explanation
PIT_NAME	The ASCII name of the PIT or PITG entry whose values are to be retrieved.
PARM_ID	The index of the parameter ID from the table below that corresponds to the parameter of interest.
PARM_VAL	The returned value of the parameter if the entry name and parameter ID are both valid.
RET_STAT	The returned error status for the routine. If any problem was encountered attempting to get the value, a non-zero value is returned.

Purpose

To obtain the value of the given PIT or PITG parameter. The PIT or PITG is specified by the ASCII name of the entry. The parameter is specified by the index of the corresponding parameter ID. The parameter ID is defined as follows:

<u>INDEX</u>	<u>PIT PARAMETER ID</u>
01	CYCLE
02	TIMER
03	SLOWDOWN

If the entry name is not found or if the parameter is invalid then the value returned by the procedure is undefined.

Continued on next page

8.4.2

CL Interface to NDT and PIT Structures, Continued

CLM\$Put_PIT_ Parameter

```
CALL CLM$Put_PIT_Parameter ( NDT_NAME  : IN STRING;  
                             PARM_ID    : IN NUMBER;  
                             ARM_VAL    : OUT NUMBER;  
                             RET_STAT   : OUT NUMBER  
                             ) ; BACKGROUND ;
```

Parameters

CLM\$Put_PIT_Parameter Parameters

Parameter	Explanation
PIT_NAME	The ASCII name of the PIT or PITG entry whose values are to be changed.
PARM_ID	The index of the parameter ID from the table below that corresponds to the parameter of interest.
PARM_VAL	The value to be placed into the parameter if the entry name and parameter ID are both valid.
RET_STAT	The returned error status for the routine. If any problem was encountered attempting to store the value, a non-zero value is returned.

Purpose

To change the value of the given PIT or PITG parameter. The PIT or PITG entry is specified by the ASCII name of the entry. The parameter is specified by the index of the corresponding parameter ID. The parameter ID is defined as follows:

<u>INDEX</u>	<u>PIT PARAMETER ID</u>
01	CYCLE
02	TIMER
03	SLOWDOWN

If the entry name is not found or if the parameter is invalid, the value returned by the procedure is undefined.

Important! If this call is used to change the cycle of the PIT, it should be immediately followed in the CL by another call to set the timer to zero. This allows for immediate execution of the PIT on the new cycle.

8.4.3 CL Interface to Image Point Demand Processing

Image point I/O processing

Programs may demand input and output processing of an image point. To facilitate this processing, the CLM image point must be built with the custom data segment parameters defined below.

These CDS parameters are used to initiate input and output processing.

Parameters

These parameters are supplied in the separate CDS package CLM_DMD3.CL. They are included in individual processes where required by Honeywell-supplied software.

PARAMETER CLM_OUT	: LOGICAL
PARAMETER CLM_PPS	: LOGICAL
PARAMETER CLM_PORT	: STRING
PARAMETER CLM_DEV	: STRING
PARAMETER CLM_PIT	: STRING
PARAMETER HDW_REF	: NUMBER(1..6)

CL Output processing

To initiate output processing, a store to the CDS parameter CLM_OUT must be made. This causes the output of all changed output parameters in the PITE entry that are also configured with the ECP option. Parameters are configured for program demand output change detection with the following entry in the PITE:

Pnn_OFNC:I/[ECP:L3/Y]

The program should store the logical value TRUE to force the action. The CDS parameter is set to FALSE when the action is complete.

CL PPS processing

To force the PPS parameters to process, a store to the CDS parameter CLM_PPS must be made. Like the operator initiated PPS processing, all parameters configured with the PPS option are input or output. A parameter is configured for PPS processing with one of the following PITE entry options:

Pnn_IFNC:I/[PPS:L4/Y]	- input PPS processing
Pnn_OFNC:I/[PPS:L4/Y]	- output PPS processing

The program should store the logical value TRUE to force the action. The CDS parameter is set to FALSE when the action is complete.

Section 9 - CLM Redundancy/Automatic Failover

9.1 Introduction

APP/CLM	Redundancy / Failover is a function supported by the TPS and Experion nodes. APP node redundancy is not available and is, therefore, not supported by APP/CLM.
CLM Redundancy	The CLM redundancy function provides an automatic switchover capability based on the redundant Application Module. The primary CLM node is on-line and communications are available for full operation. The backup CLM node is alive and in the “waiting” mode. The interface board in the backup node is loaded with the application software, but has not yet been issued a “GO” command. When a switchover occurs, the backup node immediately becomes the new primary node. The interface board is issued a “GO” command during the switchover process and it immediately begins the synchronization process between the CLM and the device network. The time it takes to complete the switchover process will vary depending on the size and speed of the device network, but normally within 4-5 seconds.
Mandatory Switchover	Switchover of the CLM node WILL occur when one of the following actions occurs: <ul style="list-style-type: none">• Primary node failure - If the primary CLM node fails due to a hardware or software error, switchover will occur.• Node power down - Switchover will occur if the operator powers down the CLM primary node.• Node reset - Switchover will occur if the operator presses the reset button on the front of the primary CLM node.• Operator shutdown - Switchover will occur if the operator issues a software “Shutdown” command from the User Station.
Configurable Switchover	Switchover of the CLM node WILL occur when ALL of the following occur: <ul style="list-style-type: none">• The CLMI, Port, or Device is configured for Failover. OPT : I / [FAILOVER=L4/Y]• The FAIL_AT field is set to zero or a positive value (time in seconds)• The backup AM/CLM node is alive and loaded.• A communications error occurs between the CLM and the configured device that is not re-established within the FAIL_AT time frame. <p>The backup AM/CLM node MUST be loaded and running for the configurable switchover to occur. If not, the device will change to the hard-fail state and a journal message will be issued.</p>
Synchronization	Upon a switchover occurrence, the interface board in the new primary node must re-establish a synchronization between the device network data base and the CLM image point data base. All parameters that have their SYN=L5/Y options in the IFNC and OFNC fields will be updated by requests to the device network for the latest data. During this process, the device will show in the SYNC state on the CLM Detail Display. As each device completes its parameter updates, the state will be changed to IN_SERV, indicating that the synchronization process is complete.
AxM in redundant CLM	The AxM node may be used as the base node for a CLM node. However, since the AxM is not redundant, the redundant CLM may not be based on AxM.

9.2 Configuration/Installation of CLM Redundancy/Automatic Failover

The AM redundancy option is a prerequisite for the implementation of CLM redundancy.

There is no difference between configuration/installation of the non-redundant CLM and redundant CLM. It is important to note that the redundant CLM always uses the lower node number for file identification. For example, the ZC file for CLM nodes 49/50 will be CLMI049.ZC and the check point file for CLM nodes 49/50 will be CLM049.CP. When configuring the NCF for the CLM nodes, configure as a standard redundant AM.

Important: The NCF configuration for the redundant AM nodes must include a minimum of 10 Extra Redundant Buffers.

9.3 CLM Redundancy

This software option requires a hardware upgrade.

A secondary CLM backs up the operation of the primary CLM. All of the application data in the primary CLM is transferred to the secondary CLM, where an exact copy of the data is maintained to be used if the primary CLM should become inoperative.

This option can be installed only in enhanced CLMs with the 68020 microprocessor (HMPU/AMR) or K4LCN-based Redundant CLMs (K4LCN/EAMR).

Failover flow

The following describes the failover scenario when a primary node is lost due to shutdown or failure.

- Primary CLM is lost (from an error or from device failover configuration)
- CLM redundancy initiates failover to the backup CLM.
- The new primary node initiates the CLM startup task.
- The CLM Interface board on the new primary (already loaded on secondary startup) is started.
- The CLM Interface board synchronizes its database with the devices in the remote network (if the board is In Service).
- Data collection is initiated.

Caution

If the CLM Configuration Builder is in use (i.e. on display at any U.S.) when the failover occurs, the CLM network memory database is reloaded from the CLM checkpoint file on the configured media. **All entry additions/modifications made since the last SAVE DATA will be lost!**

Continued on next page

9.3 CLM Redundancy, continued

CLM/AM Redundancy Information References

Application Module Implementation Guidelines in the *Implementation/Application Module - 1* binder – Provides guidelines for the AM Redundancy option.

Engineer's Reference Manual – Section 22 lists the processor status data point (PSDP) parameters related to AM Redundancy and defines the meaning of the values they contain. Section 28 details AM Redundancy.

LCN Guidelines - Implementation, Troubleshooting, and Service in the *LCN Site Planning & Installation* binder – Instructions for adding a new node to an operating LCN are under 6.10 in the *LCN Guidelines*.

LCN System Installation Manual in the *LCN Site Planning & Installation* binder – Provides installation instructions for the AM Redundancy hardware.

Five/Ten-Slot Module Service in the *Service* binder – Provides service instructions including board locations and spare parts information.

Network Data Entry in the Implementation/Startup & Reconfiguration - 1 binder – Provides instructions for on-line reconfiguration, which is needed to install the new configuration on the network.

9.4 Build NDT Entry for CLM Redundancy/Automatic Failover

When a CLM interface board, port or device is configured for failover and the CLM identifies that the configured component has failed,

- ⇒ Communications is lost,
- ⇒ FAIL_TM has expired,
- ⇒ Backup CLM is confirmed alive and ready

then failover is initiated.

When building the board, port, or device entry there is an option for failover. Set “Y” for NDT failover, or “N” to not failover and FAIL_TM to the desired time to go to state fail from a status fail.

EXAMPLE:

FAILOVER = Y

FAIL_TM = 2M

If there is a status failure on this component that does not, or is not corrected within 2 minutes, the state goes to FAIL, which constitutes a hard failure. At this point, switchover action is initiated.

9.5 Performance Considerations for Off-Node Requests to Redundant CLMs

Performance Indications

Due to the combined load of normal AM processing and the CLM-specific code executed by the change rate from the third-party network, the AM/CLM node tends to become very busy at certain times. When you combine this normal processing with additional data request that may be required from graphics on one or more GUS/US stations, there may be instances when this increased activity will cause an overload of the AM process communications queues that handle data requests.

Symptoms

The symptoms of this are the inability to access any data from the AM/CLM, yet the System Status Display shows a status value of OK. Attempting to access parameter data from the console will generally result in a “**Parameter Unavailable**” error message. In some cases, the indications may be only a system slowdown and a significant number of “**Job Queue Full**” messages appearing in the System Error Journal.

In many cases, these symptoms occur when a device on the CLM network is placed logically In Service, or a device failure and return from alarm, due to the heavy parameter store load from the synchronization process.

There is a configuration option available for the AM/CLM through the LCN NODES option of the NCF configuration pages that will allow for increased queue utilization. This allows the user to configure additional queues to handle this potential increased loading of the CLM processes.



This procedure is not a requirement and should only be done if you experience the above symptoms. If these symptoms do occur consistently, or only on occasion, follow the steps in the procedure below.

PROCEDURE – INCREASE AM COMMUNICATION QUEUE SIZE

- 1) Verify that the directory &CSD exists. If not, create the directory on the HM on an available user volume.
- 2) Use the EDITOR to create a file **APCFGnnn.X** (where *nnn* = the 3-digit AM/CLM node number; i.e. 045) in the &CSD directory. Edit the file to add the following command lines.

Important! Use spaces as delimiters between the fields and allow for no additional lines, either blank or otherwise. Node startup errors will result if this file is not carefully edited.

```
ADD_QUEUE 30          { May also be 10, 20, 40, 50 }
TOT_QUE_SZ 240        { 80, 160, 320, 400 (i.e. 8 times queue size) }
END
```

Note: If memory is a consideration, these values may be adjusted up or down to a size that will work for your specific system.

Continued on next page

9.5 Performance Considerations for Off-Node Requests to Redundant CLMs, continued

- 3) If this is a non-redundant node, skip to step 5).
- 4) If you are on LCN release R611 or later, a configuration file must be created for the backup node of the redundant pair also. This file must be identical to the file created in step 2) for the primary node. To create this file, simply copy the file of the primary node to a new file using the backup node number.

For example, node pair 45/46, the primary file APCFG045.X has already been created in step 2). To create the backup file, do the following command:

```
COPY NET>&CSD>APCFG045.X NET>&CSD>APCFG046.X
```

- 5) Modify the NCF for the AM/CLM node to pick up the configuration changes.
 - a) From the LCN NODES function of the Engineering Personality, select the AM/CLM node.
 - b) Select the MODIFY NODE target and move forward to page 3.
 - c) Select YES for the FURTHER EXTERNAL DIRECTIVES? field.
 - d) Check (CTL-F1) and Install (CTL-F2) the modified NCF.

Remember!	Be sure to set your paths for the backup NCF.
------------------	---

Redundant Nodes!	The redundant pair node should be set by the Configurator when the primary node configuration is set.
-------------------------	---

- 6) Reload the AM/CLM. If redundant, then both nodes of the redundant pair must be taken down together and reloaded.
-

Appendix A –Terminology/Definitions

A.1 Introduction

This appendix

This appendix briefly defines terms frequently used within this document and includes other pertinent reference material.

Terms used in this manual

The special terms used in this manual are defined below.

A.2 Terminology

Alarms

Two types of alarm processing are performed by the CLM

- System Alarms
- Point Alarms or Process Alarms

System alarms refer to the status of the network devices and the communications with these devices. The CLM Network Status display can be used to display the device and network status. Device failures are also output to the status notification journal. On R400 (and later) TPNs, device failures are also annunciated at the STATUS DETAIL level of the AM System Status display and by the System Status LED on the Operator Console of the User Station (US). In this document, system alarms may also be referred to as communication alarms.

Point or process alarms refer to the status of the functions performed or monitored by the system. The status of these functions is maintained in the point database of the system.

Builder Input File (BIF)

The BIF may contain all of the CLM interface structure definition entries required to establish the CLM interface between the LCN and the non-TPN network. In its completed form, the BIF represents source input to the CLM Configuration Builder, using the file input mode. Refer to the CLM Configuration Builder section for additional information.

Communication Link Module (CLM)

CLM is defined as a TPN Application Module (AM or AxM) with at least one CLM Interface board and the CLM software package installed. The CLM requires a minimum of 6-megaword memory, TPN Release 4xx (or later) revision, and Experion Release (Experion PKS 400/410).

Continued on next page

A.2 Terminology, Continued

CLM Image Point	The CLM image point is a TPN based data point (usually an AM type) that is used to emulate the non-TPN point or data address. This point is the basic mechanism that makes the non-TPN network data point appear to be directly connected to the TPN. The CLM image point emulates its corresponding non-TPN network data point, through the CLM software and point interface structure definitions. The CLM image point is created through the Data Entity Builder in the TPN Engineering Personality.
Non- TPN network	A non-TPN network and its associated devices is connected to the TPN through the CLM.
Network Definition Table (NDT)	The NDT contains the definition of every network component within the CLM realm, including: the CLM Interface board(s), the ports on each interface board, and each addressable component in the non-TPN network (i.e., gateway, bus, device, subdevice). In addition, the NDT contains the relationship between these network components.
Network device	Network Device refers to the non-TPN device or network that is interfaced to the TPN through the CLM.
Network memory	Network memory is a segment of the CLM Node's redundant memory allocated for shared access between the CLM and the interface board to facilitate CLM network communications processing. Network Memory includes the definition of the non-TPN network and message queues used by both the CLM and the interface board.
Point Interface Table (PIT)	The Point Interface Table contains the relationships between CLM image points and their corresponding actual non-TPN network points, along with conversion and data acquisition information. Each image point in a PIT is represented by a PITE. Each PIT defines a set of points grouped together, based on both processing time and physical device address. Each PIT must be related to the NDT that defines the addressable network component or device where the set of points is located. A PIT can also optionally be a member in a PITG.

Continued on next page

A.2 Terminology, Continued

Point Interface Table Entry (PITE)

The PITE defines one image point within a PIT. The PITE consists of parameters that define the interface input and output requirements between the CLM image point and the non-TPN network point. Each PITE within a PIT must have the same processing cycle rate and the same physical device address, although the specific point or register is unique to that PITE..

Point Interface Table Group (PITG)

A PITG optionally defines a set of PITs to be processed as a group. This allows “same-time” processing of PITs across different devices. A PITG represents the highest level entry in the overall point interface structure and, if used, must be defined prior to the definition of member PITs within the group. A PITG may not cross ports.

Template Definition File (TDF)

A TDF provides a complete definition for one type of CLM interface structure. Each template is actually a separate file resident on the LCN (file.zt). The set of template files that encompasses the different types of CLM application interface structures is utilized by the CLM Configuration Builder. This interface structure build process determines format, constants for the entry type, and required user inputs with corresponding validation criteria.

A.3 CLM Vital Statistics

Configuration maximums

A maximum of 20 CLMs can be configured within your overall TPN operating environment as required to support the total non-TPN network requirements of your installation.

The CLM Configuration Builder supports certain maximums for the configuration of one CLM (within the limits of the available AM memory for the CLM Node).

Table A.3-1 shows the maximum supported number of various items and files.

Configuration maximums, continued

Table A.3-1 Configuration Maximums

Maximum	Description	Per
2	CLM Interface Boards	CLM
4	Ports	Board
400	Components/Devices	Port/Network
Note: If > 1 Port, maximum is less, per next item below)		
398	Components/Devices (less board and 1 port)	CLM
400	Point Interface Tables	CLM
50	Point Interface Tables	PITG
300	PIT Entries (PITE)	PIT

Note: A single CLM operating environment supports a total of 400 network components. The total includes 1 or 2 interface boards, each with 1 to 4 ports. Because both the boards and their respective ports represent addressable network components, a maximum single CLM configuration would contain the following:

- 2 CLM interface Boards
- 8 Ports (4 ports on each board)
- 390 Network components/devices
- 400 Total components/devices (Including boards and ports)

Performance characteristics

The actual performance of a CLM is application and protocol specific. However, the platform is designed to support the following estimated design goals. Final performance values are generated based on data acquired during the development and implementation of this package.

A CLM (4-megaword memory) supports approximately 2000 points. Approximately 4000 points are supported with the larger 6- or 8-megaword memory AM.

Continued on next page

A.3 CLM Vital Statistics, Continued

CPU utilization

A 68020-based CLM, processing 90 points per second, is characterized with the following approximate performance characteristics (all percentages relate to CPU utilization):

5% system overhead
35% point processing
4% alarm processing
<hr/>
+ 20% external data access
64% total base load

This indicates that the CPU is approximately 36% available. It is expected the CLM platform software will utilize no more than 15%, leaving 21% for parameter data access. This should permit input and output of approximately 350 parameters per second, at 0.06% per parameter.

Therefore, approximate total utilization is as follows:

64% total base load
15% CLM platform software
<hr/>
+ 21% data access (350 parameters per second)
100% AM/CLM CPU Utilization

K4LCN

No timing or performance studies have been done at this time using the K4LCN AM processor board. Expectations are that CLM parameters per second will double and CPU usage will remain approximately the same or less.

Appendix B - CLM Platform Release Contents

B.1 Introduction

Media installation

The installation procedure in the appropriate installation manual (CLM or APP/CLM) covers the use of the release media to install the CLM Platform software.

The following information identifies the contents of the CLM Platform release media which has the volume name "&CLM".

Note: Each application, such as Modbus, that uses the CLM Platform also identifies the contents of the release for that application.

B.2 Release Media Contents

Table B-1 CLM Platform Release Media Contents

HM NET	Directory	File Types	Directory	File Types	Directory	File Types
	&CUS	ZC,ZI,ZT,ZS, S, XX	&CLX	SF	UCLM	AO, CL, DF, DS, EB,EC, EL
Extension	File Type			Extension	File Type	
AO*	Control Language (CL) Object Modules			S	CLMI Load Module	
CL*	CL - Custom Data Segments (CDS)			SF*	Include File for CL Interface	
DF*	Specific Variable Data Area			ZC	CLMI Configuration Modification Sample	
DS*	Custom Display Schematics (Source)			ZI	Builder Input File (BIF) Samples	
EB*	Point Build Files			ZS	Status Display Files (CLMI/Port/device/etc.)	
EC*	Executable Modules -Install,Link/Unlink CL			ZT	Builder Template Definition Files (TDFs)	
EL*	Picture Editor Multiple Compile				> Network Definition Table (NDT) Templates	
LO*	CLM Backplane			XX*	CLM Release Verification	
* These file types may be LCN Release dependent.. They are supplied for each supported release if required.						
Install CLM Platform to HM		CLMI/CCP Load Mod.		Release Install/Upgrade		
EC	LO*	S	SF*	XX*	EC*	
INSTALL	CLM	CLMI,CCP,CCPL	CLM	RLS400	U240_340	
				RLS500	U250_350	
				RLS600	U250_360	
Configuration Files				Platform Display Point Build Files		
	ZC	ZT	ZI	ZS	EB*	CL/CDS
CLMI	CLMINNN	CLMI	CLMI	CLMI	CLM_NN	NETCDS NETCDS1 NETCDS2
PITG		PITG	PITG			NETCDS3 NETCDS4
PIT		PIT	PIT		BUILD_NN	BUILD_CDS BLD2CDS CLMDCDS
		GENDISP	GENDISP		GENAPPL	GENRLCDS
Displays	DS*	DS*	DS*	DS*	DS*	DS*
	BUILDER	CLMDETD1	CLMIDSCH	CLMPITE1	CLMSCHEM	
DF*	BUILDER0	CLMDETD2	CLMLIST1	CLMPITE2	CLMSTAT1	
CLMDDB	BUILDER1	CLMDETD3	CLMLIST2	CLMPITE3	CLMID1(subp)	
	BUILDER2	CLMDETD4	CLMLIST3	CLMPITE4	CLMSTAT2	
EL*	BUILDER3	CLMERROR	CLMLIST4	CLMSBUF1	CLMSTAT3	
CLM_DS	BUILDER4	CLMGRPD1	CLMPIT1	CLMSBUF2	GENDISPA	
	BUILDH01	CLMGRPD2	CLMPIT2	CLMSBUF3	GENDISPB	
	CLMCONF	CLMGRPD3	CLMPIT3	CLMSBUF4	GENDISPC	
		CLMGRPD4	CLMPIT4		GENENTER	
CLM Platform Point Type Files		CL/CDS	CL/CDS	AO*	EC	EB*
3-State Digital Composite/Output		DIG3CDS	DIG3CDS1	DIG3CL	DIG3	DIG3L,DIG3P,DOUT3L,DOUT3P
4-State Digital Composite/Output		DIG4CDS	DIG4CDS1	DIG4CL	DIG4	DIG4L,DIG4P,DOUT4L,DOUT4P
5-State Digital Composite/Output		DIG5CDS	DIG5CDS1	DIG5CL	DIG5	DIG5L,DIG5P,DOUT5L,DOUT5P
3-State Digital Input		DIN3CDS		DIN3CL	DIN3	DIN3
4-State Digital Input		DIN4CDS		DIN4CL	DIN4	DIN4
		CLM_DMD3				
Point Type	Desc.	AM Point	Point Type	Desc.	AM Point	
DIN3	Digital Input - 3 state	Switch	FLAG	Digital Input / Output	Flag	
DIN4	Digital Input - 4 state	Switch	NUMERIC	Register Input/Outout	Numeric	
D3	Digital Composite - 3 state	Switch	PULSE INPUT	Totalizer (Accum.)	Regulatory	
D4	Digital Composite - 4 state	Switch				
D5	Digital Composite - 5 state	Switch				

end

Appendix C - CLM Startup

C.1 Automatic Startup and Initialization

The Custom Loader

The startup of each CLM Node occurs whenever the AM Node is restarted. As part of the AM load process, the Custom Loader task within the AM personality loads the CLM software and upon completion of the AM startup starts the CLM Startup task.

NOTE: The CLM Startup task runs only once per CLM startup or restart.

Automatic startup

The CLM Startup task performs the following functions automatically:

- Loads CLM Network Memory from the checkpoint file on the designated media.
Note: If a checkpoint file does not exist on the configured media or the NULL LOAD configuration option is set, Network Memory is set to an initial state. This state requires the CLM Configuration Builder process to establish CLM interface structures in Network Memory.
 - CLM on AM node
 - Forces a reset of the CLM interface board(s).
 - Downline loads configured software to the Interface board(s)*.
 - Commands the execution of the board application software*.
* Only if the board NDT operating state is in service in network memory.
 - APP/CLM on APP node
 - Forces a reset of the APP/CLM emulated interface board in the APP.
 - Loads configured software for emulated interface board.
 - Activates the execution of the configured application software*.
* Startup status is logged in the Windows Application Event Log. The Event Logs may be viewed using the Event Viewer
 - Starts all other CLM Platform tasks, including the CLM Synchronization task.
-

Synchronization

CLM Synchronization performs database synchronization between the CLM image points and the external device data as follows:

- All devices that have their operating state defined as IN_SERV in their NDT interface structure in Network Memory, are placed in the SYNC state.
- At successful completion of synchronization each device is placed in the IN_SERV operating state. If synchronization is not completed successfully, the device remains in the SYNC state.
- The PITE interface structures define the point parameters that are used in the database synchronization process.
- Each PITE point parameter can be configured for either uploading into the image point or downloading from the image point during the synchronization process. Refer to the Pnn_IFNC and Pnn_OFNC point parameters in the PITE Template section for details.

On completion of the CLM Synchronization task the CLM Scheduler task commences operation by performing the PIT scheduling function. At this point, the CLM is in normal operation.

Appendix D - CLM Memory Access Utility

D.1 Introduction

Application

This utility is specific to the standard CLM installed on an AM with appropriate communications boards – CLMI, CCP, UCIM.

This does not apply to the APP/CLM implementation.

Installation Procedure

The CLM Platform installation includes the installation of this function . The usage of the function is not intended for normal operations and is only defined in this appendix.

This function is platform oriented and does not pertain to any specific application protocol.

Memory Access Function

The CLM Memory Access function allows the user to view all CLM memory locations; this includes the memory physically existing in the CLM processor board and the CLM Interface board(s).

The user can access and view this data by selecting the SCHEM key and entering CLMIDSCH, then following the prompts of that process. The ability to output to the screen or a file is included. Either capability requires the key lock to be in the ENGR position.

Output files cannot be read by the LCN editor. They can be printed on the LCN or transferred to another media type.

For the CLMI board Release B, or greater, of the firmware is required. For the CCP board, all releases are valid. If the process is attempted, but the firmware is not proper, a message to that effect is given.

The function does not check for a valid address, but rather leaves that responsibility to the user (i.e., if the user enters improper address for a read, a read error results).

Continued on next page

D.2 CLM Memory Access Schematic

CLM Memory Access Schematic

The CLM memory access schematic is used to access board and/or CLM memory. The user selects and enters data in a menu type format, and then selects the execute target to process the request. The schematic contains a current request status field to inform the user of successful completion of the request, or if an error occurred, an abbreviated message to help interpret the error.

Note: A CDS parameter contains an error code to aid in resolving the error condition.

If the error is caused by improper data field input, the schematic highlights the field in question.

The following describes a typical use of the function.

The user must enter the CLM identifier and the Interface board identifier. The schematic verifies that entered values are valid, and are not currently being accessed by another user at a different User Station.

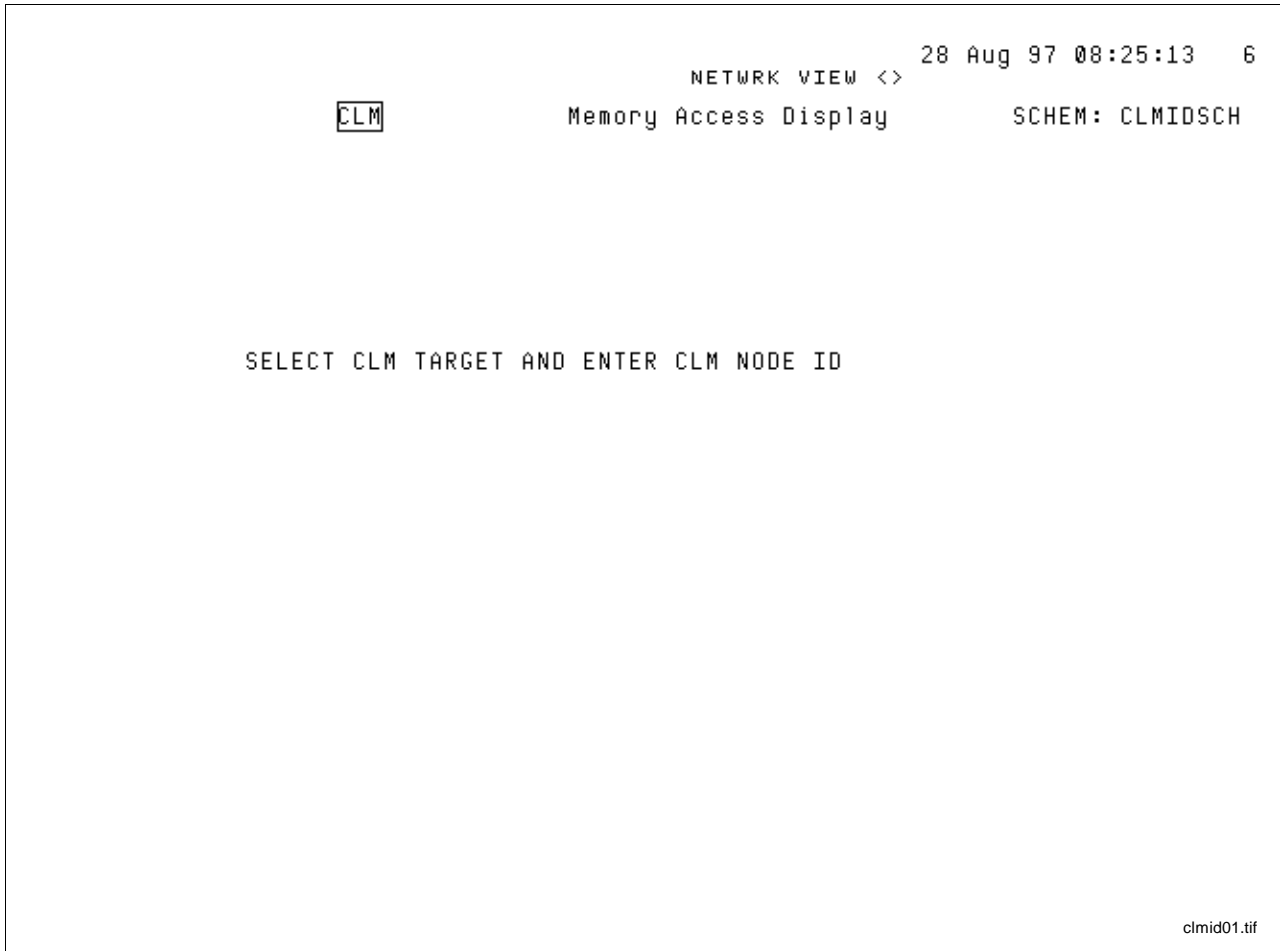
Continued on next page

D.2 CLM Memory Access Schematic, Continued

CLM Memory Access Schematic, continued

Figure D.2-1 shows the initial display. This is called up by using the SCHEM key and entering CLMIDSCH.

Figure D.2-1 CLM memory Access Display - CLM Identifier



CLM memory access display upon invocation.

Note: Entries are made by selecting the desired target, entering the data, and then pressing the enter key.

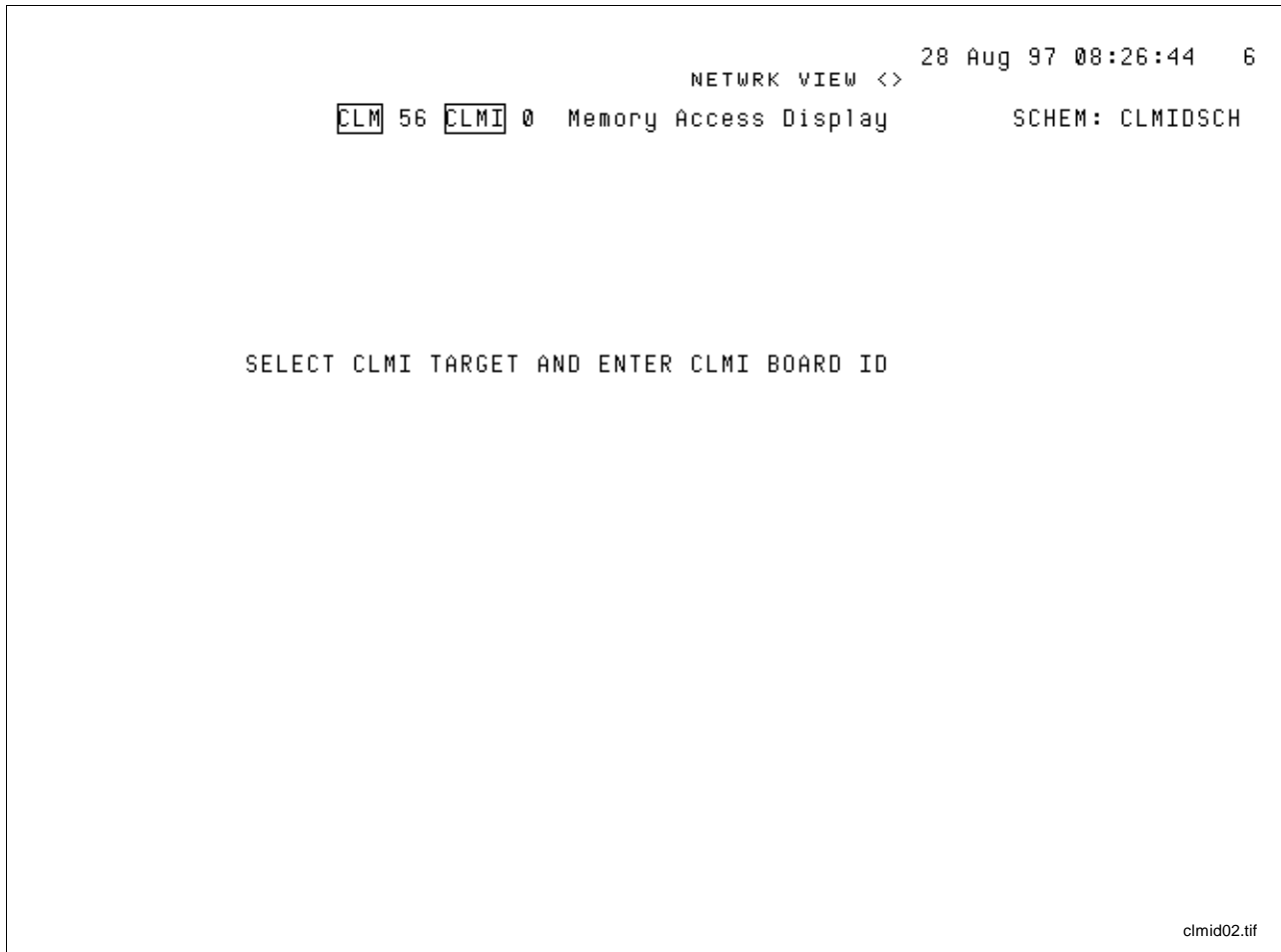
Continued on next page

D.2 CLM Memory Access Schematic, Continued

CLM Memory Access CLM Dump Schematic, continued

Enter the CLM node number. The target shows "SELECT THIS TARGET" to enter the desired board identification (1 or 2) (Refer to next screen.).

Figure D.2-2 CLM memory Access Display - CLM Board Identifier



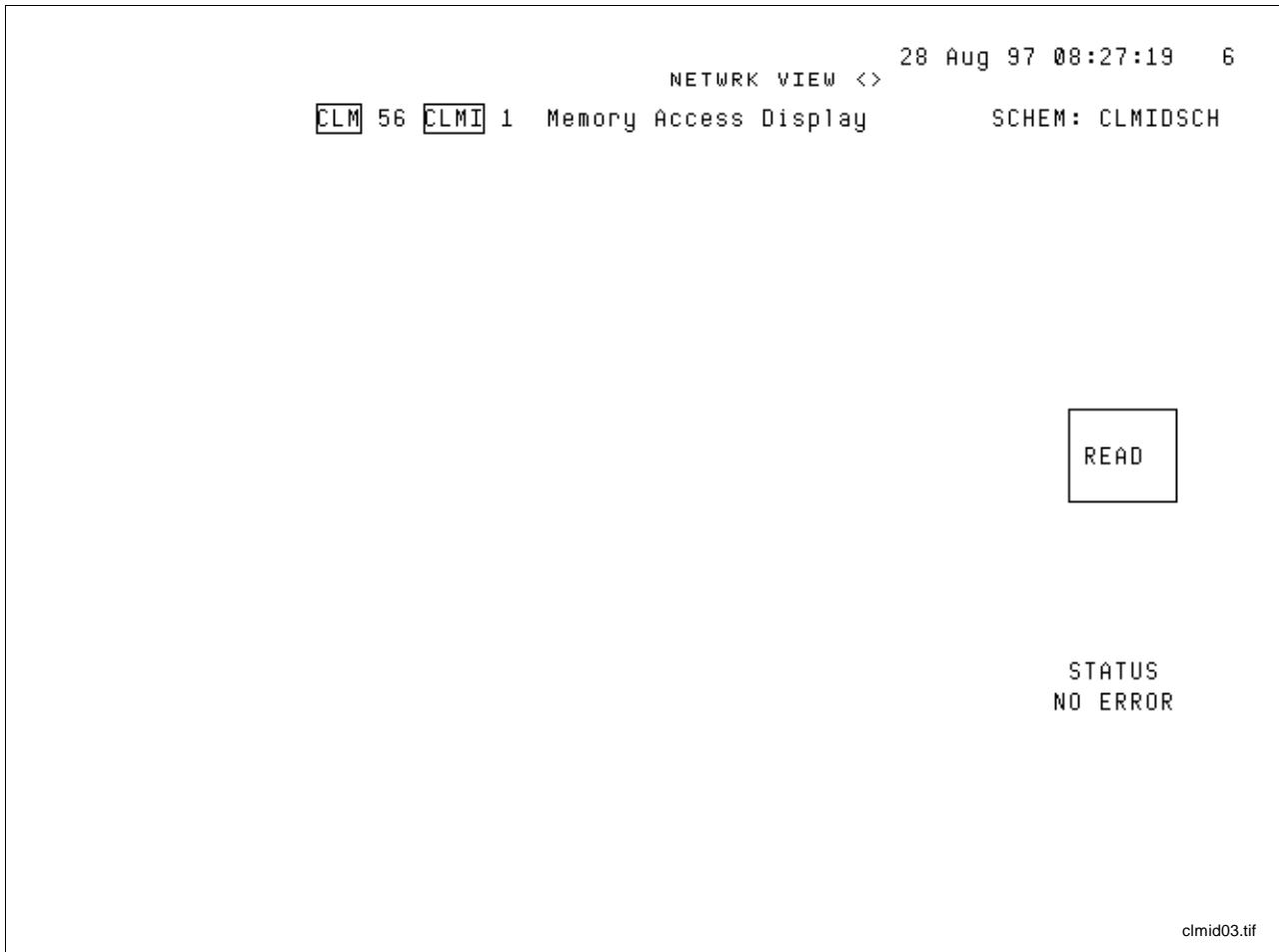
Continued on next page

D.2 CLM Memory Access Schematic, Continued

CLM Memory Access CLM Dump Schematic, continued

The target and status is shown if the CLM and board entries are valid. If the entries are not accepted, an error prompt is shown.

Figure D.2-3 CLM memory Access Display - Status



Continued on next page

D.2 CLM Memory Access Schematic, Continued

CLM Memory Access CLM Dump Schematic, continued

If the **READ** target is selected, the following display is shown.

Figure D.2-4 CLM memory Access Display - Read Display

```

28 Aug 97 08:27:53 6
NETWORK VIEW <>
CLM 56 CLMI 1 Memory Access Display SCHEM: CLMIDSCH

READ

STATUS
NO ERROR

OUTPUT TO :
SCREEN FILE
Address 1 Address 2
# Words 1 00000000 # Words 2 00000000

```

Continued on next page

D.2 CLM Memory Access Schematic, Continued

CLM Memory Access CLM Dump Schematic, continued

The user may then select if the data is to be shown on the screen or stored into a specified file. The user must also enter the starting address and the number of words desired. The display allows for entry of two starting addresses and number of words. Address one takes priority over address two. The screen display allows for twenty lines of information containing eight words of memory per line. If the user requests more than twenty lines of information, the display shows only the first twenty lines of information based on address one. The function attempts to satisfy address one information first, and if room allows, all or as much of address two as possible. If only address two is specified, address one is ignored. To aid the user in viewing simultaneous display of address one and address two information, the latter information is shown in reverse video.

Both the address and number of words must be entered to be valid. If the address is entered, but the number of words is zero, then the entry is ignored. If the address is zero, but the number of words is greater than zero, then the entry is valid and the number of words specified, starting at address zero, is displayed.

Continued on next page

D.2 CLM Memory Access Schematic, Continued

CLM Memory Access CLM Dump Schematic, continued

If the user selects the **FILE** target, an entry port appears for entry of the pathname. The user can enter the full pathname or just the filename. If only the filename is entered, the pathname defaults to NET>&CUS>filename.

Figure D.2-5 CLM memory Access Display - File Target Entry Port

The screenshot shows a terminal window with the following content:

```
                28 Aug 97 08:29:05    6
            NETWORK VIEW <>
    [CLM] 56 [CLMI] 1 Memory Access Display      SCHEM: CLMIDSCH

                                                    [READ]

                                                    STATUS
                                                    NO ERROR

OUTPUT TO :
[SCREEN] [FILE] NET>&CUS>12345678.XX
[Address 1] [Address 2]
[# Words 1] 00000000 [# Words 2] 00000000
clmid05.tif
```

The address one and address two priority methods are not valid for file usage. If address one and address two are specified, the number of words for both are valid. The file contains the specified number of words for both addresses.

If a valid address and number of words are entered, the **EXEC** target appears. The user can select targets and make changes until the **EXEC** target is selected.

Continued on next page

D.2 CLM Memory Access Schematic, Continued

CLM Memory Access CLM Dump Schematic, continued

The memory retrieved from the board or CLM is presented in the following format:
8 char address, space, word1..word8, 2 spaces, ASCII representation

Note: If the data cannot be represented by ASCII, a
"." is substituted.

The following display is shown if the user selects:

SCREEN

Address 1, then enters 50000

Words, then enters 20

EXEC

Note: MAX = FFFF

Figure D.2-6 CLM memory Access Display - Current State

The screenshot displays the CLM Memory Access Display interface. At the top right, the date and time are shown as "28 Aug 97 08:33:48" followed by a small icon. Below this, the text "NETWRK VIEW <>" is visible. The main title area shows "CLM 56 CLMI 1 Memory Access Display" and "SCHEM: CLMIDSCH". The central part of the screen contains a memory dump with three lines of data: "00050000 0000 0000 0000 0000 0000 0000 0000 0000", "00050010 0000 0000 0000 0000 0000 7F01 003E 0000>..", and "00050020 0000 0000 0000 0000". To the right of the dump are two buttons labeled "EXEC" and "READ". Below these buttons, the status is indicated as "STATUS NO ERROR". At the bottom left, the "OUTPUT TO :" section shows "SCREEN" and "FILE" as options. At the bottom right, the "Address 1" field is set to "50000", "Address 2" is empty, "# Words 1" is set to "00000020", and "# Words 2" is empty. The file name "clmid06.tif" is printed in the bottom right corner.

```
28 Aug 97 08:33:48 6
NETWRK VIEW <>
CLM 56 CLMI 1 Memory Access Display SCHEM: CLMIDSCH
00050000 0000 0000 0000 0000 0000 0000 0000 0000 .....
00050010 0000 0000 0000 0000 0000 7F01 003E 0000 .....>..
00050020 0000 0000 0000 0000 .....
EXEC
READ
STATUS
NO ERROR
OUTPUT TO :
SCREEN FILE
Address 1 50000 Address 2
# Words 1 00000020 # Words 2
clmid06.tif
```

Continued on next page

D.2 CLM Memory Access Schematic, Continued

CLM Memory Access CLM Dump Schematic, continued

The status shows the current state of the request.

The read function is an update type request. The update is halted by selecting the target again. This action stops the update, and leaves the information on the screen. If the target is selected again, the information is removed, the address/number words entries are initialized, and output to screen is defaulted.

Note: The "FAST" key is supported for retrieved values which are shown on the screen.

D.3 CLM Memory Access Error Codes

The following codes are used to identify the specific error that occurred for a request. The status field on the display indicates that the request could not be performed, and the error code contained in the CDS parameter "NUMBR1[11]" is deciphered as follows:

- | | |
|----|--|
| 0 | No Error |
| 1 | Address entered by user out of range |
| 2 | Address + (Number of words) out of range |
| 3 | File error - Unable to create file |
| 4 | File error - unable to open file |
| 5 | File error - Cannot get memory |
| 6 | Address #1 invalid |
| 7 | Address #2 invalid |
| 8 | Address error on write to CLMI board |
| 9 | Data error |
| 10 | Error initializing custom parameters |
| 11 | Error setting up custom parameters |
| 12 | Error reading from custom parameter |
| 13 | Error storing to custom parameter |
| 14 | Read error |
| 15 | Write error |
| 16 | Number of words is invalid |
| 17 | Firmware error |
| 18 | Output File is full |
| 19 | Invalid CLMI board number |
| 20 | File error - Invalid path name |
| 21 | File error - File already exists |

Note: The CLM Dump Utility function is tied to the Builder task and the Builder function has priority. Therefore, it is best that the Builder task not be executing for the node being accessed. It is possible to slow the process down for both tasks if they are running at the same time. It will work, but very slowly.

Appendix E – UCLM: CCP AND UCIM Redesigned

CCP (51402426) and UCIM (51402436) boards have been redesigned and are now known as UCLM-51306409. UCLM is a combination of CCP and UCIM.

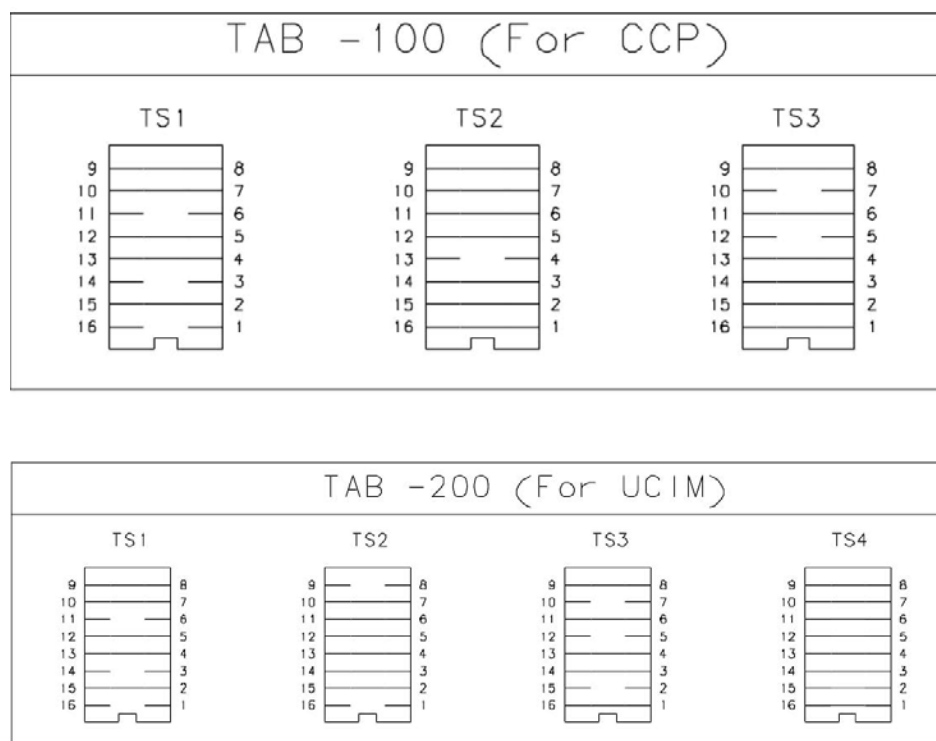
Ordering Information

To order the boards use the Honeywell part numbers as given below:

Part Number	Board Name
51306409-100	CCP
51306409-200	UCIM

E.1 Switch settings of UCLM board

Figure E-1 illustrates the redesigned CCP and UCIM settings



There is no change in the procedure to load AM personality on UCLM. Procedure remains the same as it was in CCP and UCIM.

Index

A

- Alarm of image point 2-state variables, *133*
- Alarm of image point CDS variables, *133*
- Alarming
 - Alarm count, *45, 46*
 - Deviation, *135*
 - Implementation, *137*
 - LCN, *136*
 - PIT alarm notification option, *95*
 - PITG alarm notification option, *91*
 - PV, *133, 136*
- Alarms, *326*
- alarms generated by CL, *133*
- AM, *xiv*
- AM image points, *118*
- AM Redundancy Information References, *320, 322*
- Automatic startup functions, *333*
- AxM, *317*

B

- backbuild, *34*
- Backbuild mode, *39*
- Backbuild target, *56*
- BIF, *xiv*
- BIF definition, *40*
- BIF format, *41*
- BIF source format, *40*
- BIF source statement format, *41*
- BIF usage rules, *41*
- BIF/NDT Build code example, *112*
- BIF/PIT Builder Input Example, *114*
- BIF/PITE Build code example, *116*
- BIF/PITE Builder Input Example, *116*
- BIF/PITG Builder Input Example, *113*
- Break target, *55*
- Build target, *57*
- Builder, *34*
- Builder Input File (BIF), *326*
- Building CLM Interface Structures, *70*
- BUS Template file, *110*

C

- CDS, *xiv*
- CFGERR, *140, 144*
- CHGSTATE, *146, 176, 224*
- CL, *xiv*

- CL interfaces, *299*
 - CLM\$GET_FIELD, *305*
 - CLM\$Get_NDT_Parameter, *311*
 - CLM\$Get_PIT_Parameter, *313*
 - CLM\$PUT_FIELD, *303*
 - CLM\$Put_NDT_Parameter, *312*
 - CLM\$Put_PIT_Parameter, *314*
 - CLM\$RECV_CL_MESSAGE, *308*
 - CLM\$REL_BUFFER, *302*
 - CLM\$RES_BUFFER, *301*
 - CLM\$SEND_CL_MESSAGE, *307*
 - Output processing, *315*
 - PPS processing, *315*
- CLM, *xiv*
- CLM Builder processing modes, *36*
- CLM Configuration Builder, *33, 131*
- CLM described, *19*
- CLM Image Point, *327*
- CLM implementation
 - task order of, *18*
- CLMI, *xiv*
- CLMI board and port TDF examples, *49*
- CLMI Template file, *108*
- CMDDIS, *145, 152, 156, 158, 165, 167, 182, 184, 191, 193, 201, 203, 212, 214, 231, 241, 253, 264, 275, 286*
- CMDFAIL, *145, 158, 167, 184, 193, 203, 214, 233, 243, 253, 264, 275, 286*
- Communication Link Module (CLM), *326*
- Configuration maximums, *329*
- Configuration overview, *31*
- configuration parameters, *128*
- CPU utilization, *330*
- Custom Loader, *333*

D

- Data conditioning (CL processing), *137*
- Data point types, CLM-supported
 - Counter, *133*
 - Numeric, *133*
 - Regulatory, *133*
- Database initialization or synchronization, *127*
- database synchronization, *127*
- DE, *xiv*
- DEB, *xiv*
- Delete target, *56*
- Deletions, *42*
- Demand processing, *129*

DET_STS (detailed status) parameter, *122*

Digital Outputs

CL, *143*

Digital Point Alarming, *144*

Display faceplate parameters, *136*

E

ECP, *xiv*

Entry Name target, *55*

Entry Type target., *54*

examples of Builder Type Files:, *48*

F

File Input target, *55*

four layers, description, *23*

four layers, relationships among, *24*

four logical layers, *23*

G

General form and definition of STSDF format, *123*

General forms for PITE entries, *130*

graphic overview of the Build process, *34*

Grouping network points, *92*

GUS, *xiv*

H

Hardware and software, *20*

Help, *54, 56*

HM, *xiv*

I

Image point Build operation, *127*

Image point design, *125*

Input File mode, *36*

Input/output control and schedule, *136*

Intended audience for this manual, *16*

interface structure terms and relationships, *25*

L

LCN, *xiv*

M

messages and prompts, *59*

N

NaN, *xiv*

NCF, *xiv*

NDT, *xiv*

NDT form and definition, *80–81*

NDT makeup, *80*

NDT Template Definition file examples, *108–12*

NDT Template for a PORT, *44*

NDT Template General Form, *81*

Network Definition Table, *26*

Network Definition Table (NDT), *33, 118, 327*

Network Definition Table (NDT) data structure, *43*

Network device, *327*

network interface layer, *23*

Network memory, *327*

NO_MOVE, *203, 214, 253, 264, 275, 286*

Non- TPN network, *327*

Non-Honeywell device data point type, *134–35*

O

OFF_NORM, *145, 158, 167, 176, 203, 214, 224, 253, 264*

On-Line Builder mode, *37*

Other data requirements (support data), *137*

overview of image point operation, *127*

P

Performance characteristics, *329*

PIT, *xiv, 28*

PIT explanations, *92*

PIT name target, *55*

PIT structure, *131*

PIT structure levels, *119*

PIT Template Definition file example, *114*

PIT template example, *114*

PIT Template General Form, *93*

PITE, *xiv*

PITE entry options, *129*

PITE interface structure, *47*

PITE parameter sets, *97–98*

PITE template conversion codes, *129*

PITE Template Definition File example, *115–16*

PITE template example, *115*

PITE Template General Form, *98–99*

PITG, *xiv, 28*

PITG template example, *113*

PITG parameters, *91*

PITG Template Definition file example, *113*

PITG Template General Form, *90*

PITG usage, *90*

platform layer, *23*

Point Interface Table, *28*

Point Interface Table (PIT), *26, 27, 33, 327*

Point Interface Table (PIT) interface structure, *46*

Point Interface Table (PIT) structure, *131*

Point interface Table Entry (PITE), *33*

Point Interface Table Entry (PITE), *27, 28, 29, 328*

- Point Interface Table Group, 28
- Point Interface Table Group (PITG), 27, 33, 328
- Point Interface Table Group (PITG) interface
 - structure, 45
- Point Interface Table Structure, 27
- Point Type Options
 - Analog Input Point, 134
 - Digital Input Point, 134
 - Input Point Types
 - analog input points, 134
 - custom or device specific input data points
 - calculated data points, 134
 - Smart Transmitters, 134
 - digital input points, 134
 - pulse input points, 134
 - Numeric (Analog Input) Point, 134
- Point type selection guidelines, 134
- PORT Template file, 109
- PPS, *xiv*
- PPS processing, 129
- process application layer, 23
- protocol layer, 23

R

- RB, *xiv*
- RBE, *xiv*
- required hardware and software, 20
- required task order of CLM implementation, 18

S

- Skip target, 57
- Source statement entry examples, 48

- Source statement entry examples, continued, 49
- status and error messages, 60
- Status Definition File (STSDF), 122
- STI, *xiv*
- STSDF, 122
- STSDF format, 122–23
- SUB-DEVICE Template file, 111
- suggested reading programs, 18
- Switchover, 317
- Synchronization, 317, 333

T

- targets, 54
- task order of CLM implementation, 18
- TDF, *xiv*
- TDF general rules, 79
- TDF special characters, 73
- TDFs, 35
- Template Definition File, 49
- Template Definition File (TDF), 328
- Template Definition Files, 118
- Template Definition Files (TDFs), 35
- The HELP button, 54
- Tracking**, 181, 230
- Types of templates, 72

U

- UNC_CHG, 145, 158, 167, 184, 193, 203, 214, 233, 243, 253, 264, 275, 286
- Using the Builder, 131
- UXS, *xiv*

